REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
03/07/2014	Final Technical Report	06-01-2011 to 31-05-2014

4. TITLE AND SUBTITLE
Resilient Organizational Architectures for Command and Control

5a. CONTRACT NUMBER

5b. GRANT NUMBER
N00014-11-1-0129

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Levis, Alexander H.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
System Architectures Laboratory
Dept. of Electrical and Computer Engineering MS 1G5
George Mason University
Fairfax, VA 22030

8. PERFORMING ORGANIZATION REPORT NUMBER
SAL/FR-14-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
OFFICE OF NAVAL RESEARCH
875 N. RANDOLPH STREET
ONE LIBERTY CENTER
ARLINGTON, VA 22203-1995

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release: distribution is Unlimited

20151013108

13. SUPPLEMENTARY NOTES

14. ABSTRACT
A formal definition of organizational resilience is articulated in which the three stages of avoidance, survival, and recovery are included. Three measures of performance, Capacity, Tolerance, and Flexibility are considered and several computable measures for each are defined. A case study of the resilience of a Maritime Operations Center operating in a contested cyber environment illustrate the research results. In addition, an approach for integrating and synchronizing the planning by functional components was developed. Finally, a domain specific approach was developed for using meta-modeling and multi-modeling to address the evaluation the design of C2 organizations.

15. SUBJECT TERMS
Command and Control, Resilience, Organization Design,

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPHONE NUMBER *(Include area code)*
Unclassified	Unclassified	Unclassifie d	SAR	68	Alexander H. Levis / 703 993 1619

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATE COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33315-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report. e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

SYSTEM ARCHITECTURES LABORATORY
DEPT. OF ELECTRICAL AND COMPUTER ENGINEERING
GEORGE MASON UNIVERSITY
Fairfax, VA22030

RESILIENT ORGANIZATIONAL ARCHITECTURES FOR COMMAND AND CONTROL

ONR Award No: N00014-11-1-0129

1 June 2011 – 31 May 2014

FINAL TECHNICAL REPORT

By

Alexander H. Levis

Submitted to:
Office of Naval Research
Attn: Dr. Jeffrey G. Morrison
Code 341
825 N. Randolph St., Suite 1425
Arlington, VA 22203-1995

Submitted by:
Alexander H. Levis
Principal Investigator
alevis@gmu.edu
Phone: 703 993 1619
Fax: 703 993 1601

Resilient Organizational Architectures for Command and Control

ONR Award No: N00014-11-1-0129

1 June 2011 – 31 May 2014

FINAL TECHNICAL REPORT

ABSTRACT

This is the final technical report of the contract entitled "Resilient Organizational Architectures for Command and Control" that was conducted by the System Architectures Laboratory of George Mason University between 1 June 20011 and 30 May 2014. The overall objectives of the effort were (a) to investigate resilience and relate it to structural properties of organizations engaged in command and control in a contested environment; (b) develop and apply measures that discriminate between architectures in terms of resilience and ability to scale up (augmentation) or scale down (de-augmentation); and (c) to apply the research results to a significant navy example. While the project was executed over the three year period, only the first year was funded. Consequently, the scope of the effort was reduced substantially and three distinct tasks were carried out, each leading to a PhD dissertation. In the first task, a formal definition of organizational resilience was articulated in which the three stages of avoidance, survival, and recovery were included. Loss of GPS and loss of a software application were considered as examples of operating in a contested cyber environment. Three measures of performance, Capacity, Tolerance, and Flexibility were considered and several computable measures for each were defined. A case study of the resilience of a Maritime Operations Center with and without augmentation to the loss of the mission order generation software was used to illustrate the research results. In the second task the focus was on approaches for integrated course of action development. Currently, functional component planning is often separated into multiple parallel processes with limited information sharing. Once developed, de-confliction is necessary, but this may not be completed within the time available. Integrating and synchronizing the effects of functional components is an important military principle. The new approach to integrated planning is focused on common conceptual model creation early in the planning process. A process, named Co-Design, was developed to enable through information sharing agreement on these elements in discrete steps in a logical order. The feasibility of this approach was demonstrated through a combination of planning process modeling and course of action performance modeling. Courses of action developed using Co-Design were shown to have a much greater level of integration. The third task was focused on using meta-modeling and multi-modeling to address C2 problems. The approach is domain specific: identification of the domain and the supporting modeling techniques is the first step. Then a Domain Specific Multi-Modeling Workflow Language (DSMWL), supported by a domain ontology, is developed and then used to construct workflows that capture interoperations between various models. The domain ontology provides semantic guidance to effect valid model interoperation. The approach was illustrated using a case study from the drug interdiction and intelligence domain.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

This is the final technical report of the contract entitled "Resilient Organizational Architectures for Command and Control" that was conducted by the System Architectures Laboratory of George Mason University between 1 June 20011 and 30 May 2014. The initially proposed effort was a continuation of the research on Scalable Adaptive Architectures for Maritime Operations Center Command and Control that was conducted by George Mason University between January 16, 2008 and April 30, 2011 (ONR N00014-08-1-0319). The next step in the design approach was to consider resilience and relate it to structural properties of organizations engaged in command and control in a contested environment. If this were to be achieved, then the next step would be to develop and apply measures that discriminate between architectures in terms of resilience and ability to scale up (augmentation) or scale down (de-augmentation). The results of this technical effort are reported in Section II.

A particular issue that arises when the designing and planning process is done by functional components that are geographically distributed (as in an augmented MOC) with limited information sharing. Once the functional components develop their respective contributions to the plan, de-confliction is necessary, but this may not be completed within the time available. Integrating and synchronizing the effects of functional components is an important military principle. The approach presented in Section III of this report to achieve integrated planning is focused on common conceptual model creation early in the planning process. A process, named Co-Design, was developed to enable through information sharing agreement on these elements in discrete steps in a logical order. Courses of action developed using Co-Design were shown to have a much greater level of integration.

Modeling and analysis of Command and Control architectures requires the use of multiple interoperating models. This requires the use of tools for formulating the workflow that governs the interoperation of the models. A concurrent problem is establishing the validity of the interoperation. The third research effort was focused on using meta-modeling and multi-modeling to address these problems (Section IV). The approach is domain specific: identification of the domain and the supporting modeling techniques is the first step. Then a Domain Specific Multi-Modeling Workflow Language (DSMWL), supported by a domain ontology, is developed and then used to construct workflows that capture interoperations between various models. The domain ontology provides semantic guidance to effect valid model interoperation. The approach was illustrated using a case study from the drug interdiction and intelligence domain.

While the project was executed over the three year period, only the first year was funded. Consequently, the scope of the effort was reduced substantially and three distinct tasks were carried out, each leading to a PhD dissertation.

II. RESILIENCE OF C2 ARCHITECTURES

2.1 Introduction

The word 'resilience' is derived from the Latin words 'resilire' which meant: "the ability to rebound or jump-back." The International Council on Systems Engineering (INCOSE) defines resilience as "the ability of organizational, hardware and software systems to mitigate the severity and likelihood of failures or losses, to adapt to changing conditions, and to respond appropriately after the fact" [1]. Many other highly related definitions for resilience have been developed, however all involve the following common themes: avoidance, survival, recovery, disruption. The definition of resilience from [2] as "the ability to avoid, survive and recover from disruption" will be used.

The objective of this research was to describe a quantitative approach to measuring the expected resilience of a command and control organization based on its architecture. The main idea is that resilience can be measured through its attributes, and that these measures may be combined into a holistic evaluation of resilience. To illustrate this approach, we consider the resilience of a Maritime Operations Center's (MOC) command and control system to exercise or implement a capability when a disruption occurs. This section highlights key aspects in resilience that must be considered in any evaluation. Section 2.2 describes the attributes of resilience and their measures. Section 2.3 introduces a holistic means of combining the measures. Section 2.4 contains the MOC case study. Section 2.5 presents observations and suggestions for future work.

Resilience includes the notion of disruption. INCOSE defines disruption as "the initiating event of a reduction in performance. A disruption may be either a sudden or a sustained event..." [1]. Jackson [2] defines disruptions as events which jeopardize a system's ability to perform its intended capabilities.

An evaluation of resilience must also include temporal aspects. Timescales may vary based upon the system under consideration. However, the timescale can be normalized to allow for fairer comparisons. Figure 1 illustrates the significance of time when examining resilience and is originally described in [6].

In Fig. 1, phases of resilience identified in [2] are overlaid on the time axis. The evaluation begins at some initial time, defined as time t_0. A disruption occurs at time t_d. The system reaches some minimum operating performance level at time t_{min}, and returns to a pre-disruption state at time t_{ret}. The avoidance phase of resilience runs from time t_0 to time t_d, the survival phase runs from time t_d to t_{min}, and the recovery phase runs from t_{min} to t_{ret}. The performance is evaluated using a Measure of Performance (MoP) for a single capability of the system as described by the architecture. During the avoidance phase, a system is operating at some normal operating level of capability, defined above as Value$_2$ (V_2). When a disruption occurs at time t_d, the level of capability decreases to some minimum value, V_1, at time t_{min}. The system has a minimum threshold level of capability, V_T, below which performance is deemed un-acceptable, or below which a catastrophic failure could result.

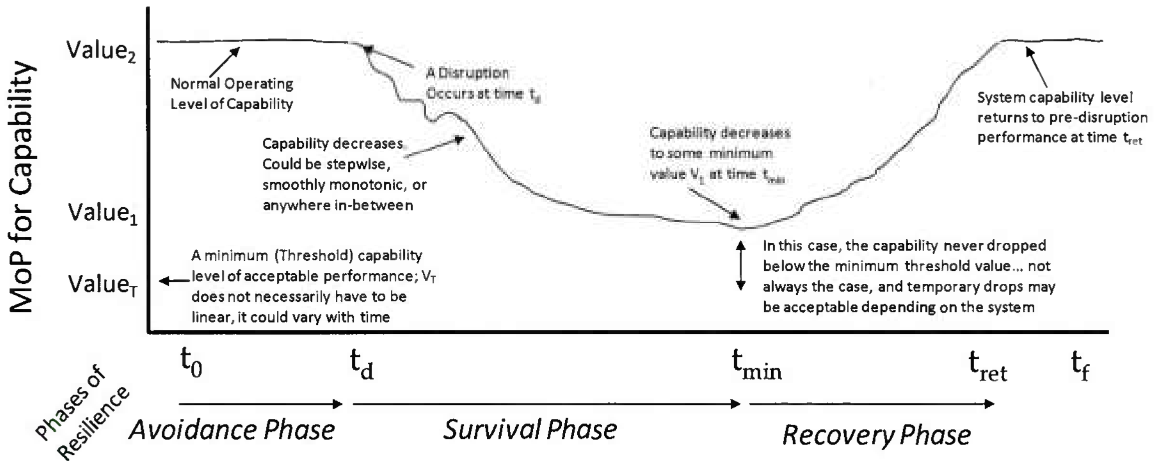


Fig. 1: Temporal Aspects in Evaluating Resilience

The approach uses the architecture of a command and control system to evaluate its resilience. Architecture is defined as "the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution" [3]. In simple terms, we can view the architecture as the high-level design of the system. Architects develop the overall design, while engineers design and deliver systems which conform to that architecture. By representing the architecture of a system in a rigorous way, one can analyze the design for key properties, and simulate the design to examine for desired performance and behavior aspects. In this manner, one can make decisions and improvements far earlier in the process, saving time, money and ultimately delivering better results.

Petri Net based architecture models are used for a number of reasons. They are rigorous (meaning that defined mathematical models underlie all aspects of Petri Net theory), visualize-able because of its graph theoretic underpinnings, and executable. These properties of Petri Nets support analyzing structural, behavioral, and performance characteristics of the architecture via simulation as well as static analyses. Finally, established and traceable means exist for translating other architectural approaches (for example Business Process Model and Notation or BPMN) into Petri Net format.

2.2 The Attributes of Resilience and their Measures

On the basis of the existing body of resilience knowledge, Jackson [2] defines four primary attributes which characterize resilience: tolerance, flexibility, capacity, and inter-element collaboration. The authors in [10] later re-describe inter-element collaboration as cohesion. This approach retains the term inter-element collaboration because cohesion is already used in [6] and [9] in a distinctly different manner. The approach described in this work partially redefines the attributes in [2] and extends them to better support the overall evaluation of resilience. Tolerance is the ability to degrade gracefully after a disruption or attack. Flexibility is the ability of a system to reorganize its elements to maintain its capabilities at degraded or even pre-disruption levels. Capacity is the ability to operate at a certain level as defined by a given measure. We further define capacity as the available capability margin between current operating levels and minimum threshold operating levels. Inter-Element Collaboration describes unplanned cooperation within a system (typically an organization) to share resources or work together in new ways. Inter-element collaboration involves the emergent properties, often human-related, of many systems and is not considered in this evaluation approach.

Tolerance is the ability to degrade gracefully after a disruption or attack. To measure graceful degradation, we consider the rate of departure (Tol_{RD}) from normal operating conditions. Rate of Departure (Tol_{RD}) is the rate of change over time in system effectiveness in meeting its requirements. This encapsulates both the temporal aspects of resilience (t_d and t_{min}), as well as the effectiveness aspects of how the system performs with respect to its requirements and how effectiveness changes during the survival phase (post disruption). Effectiveness can be measured by comparing the system performance with respect to defined Measures of Performance (MoP) against the corresponding requirements. Papers [4] and [5] describe a methodology of comparing system performance to system requirements as the intersection of the locus of performance (L_p) and the locus of requirements (L_r). System performance is characterized by the applicable MoP selected by system development team. The performance locus describes the range of system performance in the defined MoP space as the parameters of various situations are varied according to expected conditions. The requirements locus defines the required system performance levels over the same MoP space. To examine the intersection of the performance and requirements locus, a scenario is required. Parameters of interest (e.g. response time, or inter-arrival time) are varied to form a parameter locus. The executable architecture is simulated at each point in the parameter locus to determine the locus of performance. The two loci, L_p and L_r, are then depicted in a common reference frame. System effectiveness at meeting the established requirements is determined by measuring the intersection of the two loci in the common reference frame. Where the approach in [4] and [5] is static, this approach adds time. Specifically, the intersection of L_p and L_r is measured at pre-disruption (prior to t_d) and post disruption (at t_{min}) time periods, and computed using Equation (1) yielding in a change of effectiveness per unit of time. Figure 2 shows an abstract visualization of rate of departure.

Other means of measuring tolerance exist and are discussed in [6]. For example, resilient systems also typically exhibit high fault tolerance: they continue providing their main functionality despite the occurrence of one or more element-level failures. A second measure of tolerance, fault tolerance, examines the elements that can fail prior to a loss of capability using cut vertexes. A third measure of tolerance, point of failure tolerance, examines the relatedness of individual failures to a loss of overall capability. When considering faults, it is important to understand the relatedness of failures at the element level to a loss of functionality or a loss of capability; whether single element level failures tend to induce a failure of the entire system or large portions of the system.

$$Tol_{RD} = \frac{\left[\frac{L_p \cap L_r}{L_p}, t_d\right] - \left[\frac{L_p \cap L_r}{L_p}, t_{\min}\right]}{t_{\min} - t_d} \qquad (1)$$

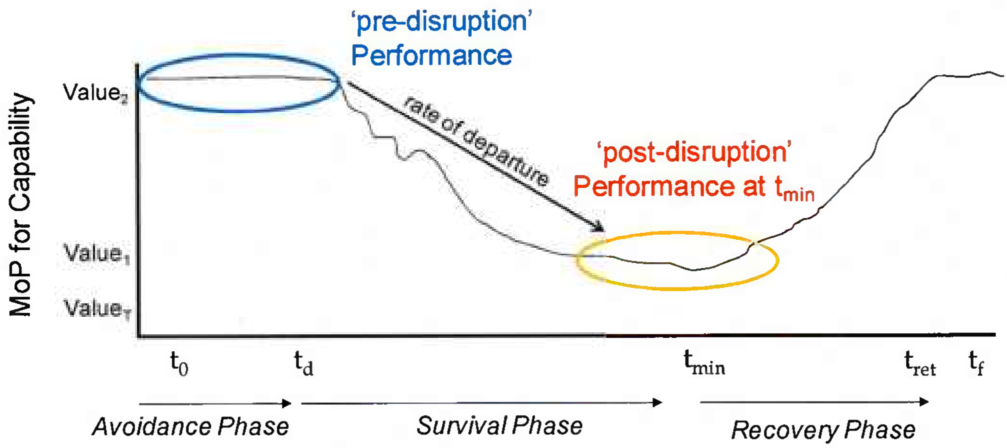


Fig. 2: Abstract Visualization of Rate of Departure

In contrast to tolerance, flexibility is the ability of a system to reorganize and adapt itself to changing conditions. Flexibility is an enabler of adjustment used by many systems to maintain their functionality during the changing conditions which follow a disruption. The graph theoretic interpretation of Petri Nets can be used to examine flexibility. Valraud and Levis [7] demonstrated the use of Petri net place-invariants to describe information flow paths and functionalities in an architecture. In their approach, a simple information flow path corresponds to a simple functionality of the system described by the architecture. A complete information flow path is obtained by coalescing all of the simple information flow paths terminating in a common sink. A complete information flow path corresponds to a complete functionality described by the architecture and defined as the partially ordered set of functions that generate a specific output. A capability is then the instantiation of one or more related complete functionalities. A well known technique to solve for the place-invariants of Petri Nets is provided in [8].

The flexibility of an architecture proposed for a certain capability can be measured by Proportion of Use. Proportion of Use (PoU) reflects the fraction of the total elements used by any given simple functionality to deliver the overall capability. For example, does the average functionality use 10% of the elements, or 80% of the elements supporting that capability? Systems with low proportion of use are more resilient to a disruption, since each element is involved in comparatively fewer simple functionalities, and easier to reorganize, because elements are less extensively used in the capability. Systems with high proportions of use are less resilient to disruption, since elements tend to be involved in comparatively more simple functionalities for a given capability, and more difficult to reorganize, because each element is extensively involved in the simple functionalities needed to deliver the overall capability. PoU implies substitutable elements. A separate measure further described in [6], fault tolerance, uses cut vertexes to indirectly examine element criticality and loss of functionality. Proportion of Use is defined in Equation (2). A second means of measuring flexibility using the graph-theoretic properties of Petri Nets is defined in [9].

$$\text{PoU} = \frac{\frac{\sum_{i=1}^{r} B_i}{E}}{r} = \frac{\sum_{i=1}^{r} B_i}{rE} \qquad (2)$$

where:

r = total number of information flow paths ℓ

B_i = number of elements e contained by path ℓ_i

E = total number of element

There are three primary means of addressing capacity when time is also considered. Buffering Capacity is the capability margin available immediately at the time of disruption or attack. Reactive Capacity accounts for the fact that certain systems are able to bring additional capacity on line after a given reaction time, defined as t_{rc}. This allows for the system to increase capacity to some maximum value, V_{max}. Given a system survives a disruption, Residual Capacity describes the remaining capacity above the threshold requirements and captures system vulnerability to a follow-on disruption that might occur in quick succession to the original disruption. Figure 3 describes how to compute each aspect of capacity when considering time. Figure 3 was originally described in [6].

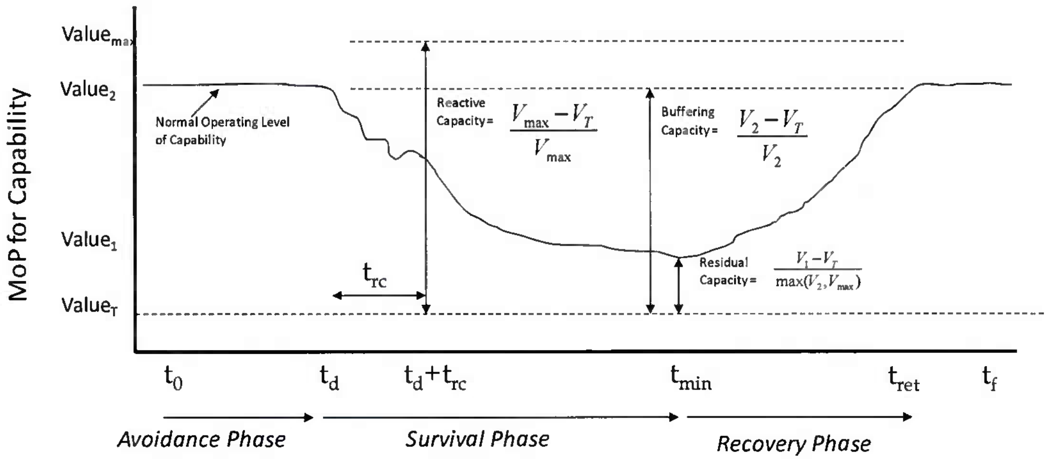


Fig. 3: Measuring Capacity

2.3 Combining the Measures to Evaluate Resilience

Section 2.2 defined measures for each attribute of resilience: capacity, tolerance, flexibility. A holistic evaluation is possible by first, selecting the appropriate metric for each attribute; second, measuring the architecture's performance against each metric; and third, comparing the architecture's performance against a required performance level for each attribute. Resilience-related improvements to the design can now be quantified and alternative architectures can be compared. The idea is to evaluate the resilience performance of the baseline architecture against the resilience requirements established by the system developers. Then either compare the baseline against alternative architectures, or make improvements to the baseline to move its performance into a desired range. (Fig. 4)

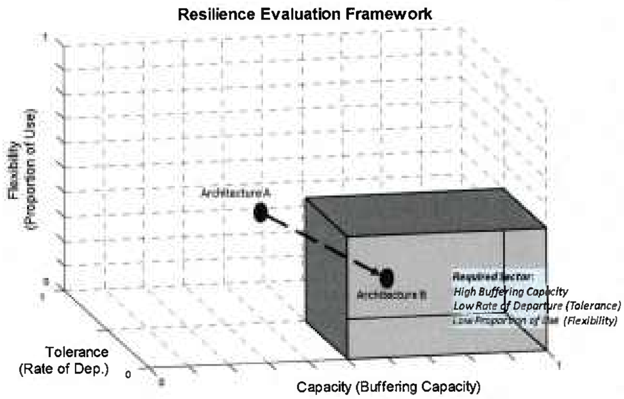


Fig. 4: Resilience Evaluation

2.4. Maritime Operations Center Organization Case Study

This case study involves the Maritime Operations Center (MOC). As the United States' presence and engagement continues on a global scale, the US Navy is transitioning portions of its command and control organizations to a MOC structure. A MOC is a large, distributed organization at the fleet level, with command and control responsibilities to "manage [routine] operations and be able to smoothly transition from peacetime operations to disaster relief operations and major combat operations, while still handling fleet management functions" [11]. The MOC is organized beneath a Joint Force Maritime Component Command (JFMCC). The MOC receives orders from the JFMCC, conducts planning operations, and generates Operations Orders (OPORD) for execution by the units assigned to the MOC. Figure 5 shows a picture of ships from the US Navy 4th Fleet undergoing training exercises during MOC certification accreditation [11]. Information regarding the MOC used in this case study is based on GMU System Architectures Laboratory (SAL) research for the Office of Naval Research (ONR) under contract number (N00014-08-1-0319). This case study used a baseline and augmented MOC model constructed by SAL staff as a foundation, made several modifications, and then applied the approach described in this research.

The MOC in this case study involves six major Decision Making (DM) organizations: Assessment, Operational Intelligence, Future Plans, Command, Current Plans, and Current Operations. These organizations work in concert to conduct command and control of Naval and Joint forces on the surface, below the surface, in the airspace and ashore.

Like many human organizations, augmentation is a typical strategy for dealing with crises and uncertainty in work load. This case study will compare two different candidate architectures for the MOC: a baseline MOC and an Augmented MOC, where the Augmented MOC adds additional nodes for Operational Intelligence and Future Plans, such that cross talk exists between nodes. These additional nodes, once called, require time to establish and are available after a given reaction time.

Fig. 5: US 4th Fleet MOC, International Exercise PANAMAX 2008

A primary capability of the MOC is to generate mission orders for subordinate unit execution, based on incoming JFMCC orders (higher HQ). The appropriate Measures of Performance (MoP) in this case is the mission orders generation rate, stated as number of mission orders generated per 24 hours, and the Average System Time from when an order from higher headquarters is received, to the time at which it is disseminated to subordinate units as an OPORD as a rate per 24 hours. Put another way, if an order takes 4 hours to process, the mission order generation rate is 6 orders per 24 hours.

Orders arrive at the MOC from the JFMCC approximately every 3.5 to 4 hours, with an execution time of 24 hours later. If the MOC spends more than 8 hours to generate mission orders for their subordinate units, then the subordinate units do not have sufficient time to conduct their own planning, move into position, and execute the mission. This is essentially an extension of the traditional 1/3:2/3 planning rule, where higher units do not take more than 1/3 of available time to ensure lower units can successfully execute the mission. Therefore, if the MOC takes longer than approximately 8 hours to generate mission orders (i.e., falls below a mission order generate rate of 3 per 24 hours), the mission is put in jeopardy because subordinate units may not be able to execute in time.

Like most operations centers, the MOC is dependent upon software to automate and improve its functioning. In this case study, we are examining the resilience of the MOC's capability to 'Generate Mission Orders' to the disruption 'loss of situational awareness software.' When a new release was received, the update caused both versions to crash, and attempts to restart were unsuccessful.

Loss of this software affects the Information Fusion stage of each decision making organization, extending the process time associated with that step. Each DM organization can still complete the Generate Mission Order process, but the process transitions to a manual backup, and requires a longer time to complete. In this case, the manual process takes 3 to 5 times as long as the software supported information fusion process. The software failure occurs at t_d = 48 hours. 24 hours are required to bring additional (augmented) capacity on-line; therefore, t_{rc} = 24 hours.

Architecture

An organizational architecture, a potential design for the MOC, is depicted in Fig. 6 (Base MOC) and Fig. 7 (Augmented MOC). Note that in the Augmented MOC an additional Operational Intelligence and additional Future Plans cells are added, with cross talk to the original cells. These figures are generated in CAESAR III. Each Decision Making (DM) organization is shown as a modified rectangle; the arcs represent fixed and variable connections (interactions) between decision making organizations by which information (or signals) is passed. Fixed connections between decision nodes indicate interactions which do not vary, whereas variable connections may change between situations.

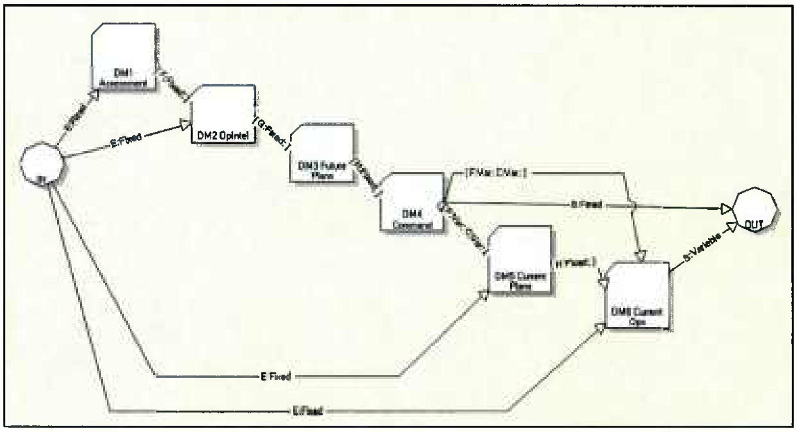


Fig. 6: The Base MOC Organizational Design

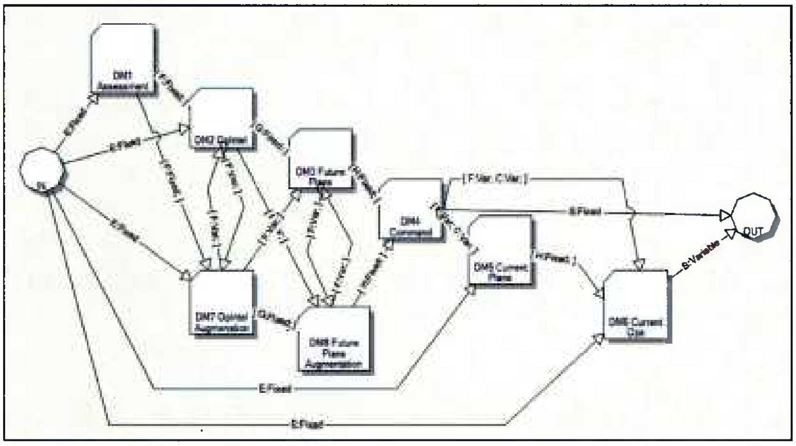


Fig. 7: The Augmented MOC Organizational Design

Remy et a. [12] introduced a four stage (later expanded to a five stage) interacting decision maker model based upon Petri Net Theory and the lattice algorithm. Each DM organization is modeled using the five stage decision maker model, therefore each DM organization shown as a rectangle in Figs. 6 and 7 can be mathematically described using a Petri net with interactions defined in Remy et al. [12]. See Fig.8 from Kansal et al. [13].

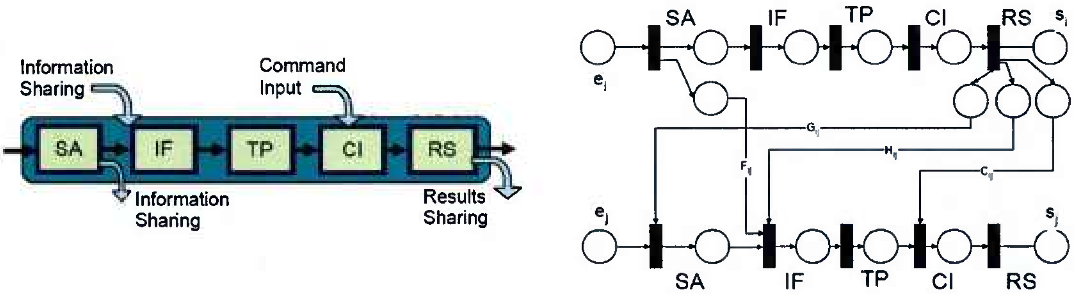


Fig. 8: Five Stage Model of Each DM Node

The individual DM nodes receive either a signal from the external environment or from another DM node. “The Situation Assessment (SA) stage represents the processing of the incoming signal to obtain the assessed situation that may be shared with other DMs. The decision maker can also receive situation assessment signals from other decision makers within the organization; these signals are then fused together in the Information Fusion (IF) stage to produce the fused situation assessment. The fused information is then processed at the Task Processing (TP) stage to produce a signal that contains the task information necessary to select a response. Command input from superiors is also received. The Command Interpretation (CI) stage then combines internal and external guidance to produce the input to the Response Selection (RS) stage. The RS stage then produces the output to the environment or to other organization members.” [13].Using the theory outlined in Remy et al. [12], CAESAR III uses the Lattice algorithm to generate feasible solutions that represent all possible architectures that meet a set of defined constraints. These solutions are represented as Ordinary Petri Nets. Figure 9 is a Petri Net representation of the DM organization shown in Fig. 7.

In this case, the primary output of the Generate Mission Orders capability is a mission order. The places P53 and P55 shown in Fig. 9 are the primary components of that mission order, where T5 is the transmission of that mission order to subordinate units. For example, the primary output of the future ops cell is an OPORD, corresponding to P53. The primary output of the current ops cell is a Fragmentary Order (FRAGORD) situation report and a synchronization matrix, corresponding to P55. A subordinate unit will execute the mission when either or both components are present, however, it will not execute if neither is present.

Using the CAESAR III generated Petri net, we can next add further necessary logic to the net and instrument it to support simulation. Care is taken to ensure that changes do not affect the overall structural properties of the original net (for example to change the nature of the information flow paths). Time was added to the original Petri Net and appropriate stochastic delays estimated for each step to represent the amount of time required for each task. The arcs were inscribed to ensure a single incoming mission order from a higher headquarters is matched up correctly as different organizations within the MOC perform their roles (i.e. when the OPORD is approved in the Command cell, that it matches the Synch Matrix and FRAGORD from the Current Operations cell.) The resulting Petri net is shown in Fig.10.

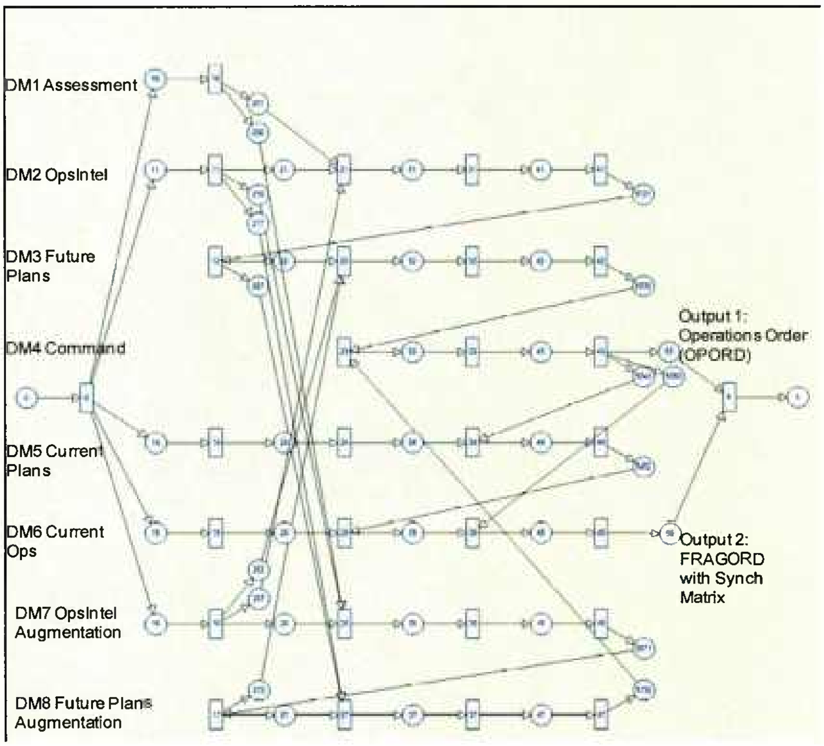


Fig. 9: The Augmented MOC Universal Net in Petri Net Form

MOC Case Study Results

Once the architecture is developed, sufficiently verified, and any errors / revisions are addressed, it may be used to support the analyses described in Section 3. To demonstrate fully the approach developed in this research, all measures of capacity, tolerance, and flexibility are calculated. However, an architect with the overall development team could in principle investigate only those measures of special interest. More generally, it is not necessary to calculate all measures if the architect and development team know a priori which measures are of greatest interest.

A. *Capacity*

Returning to our method for calculating capacity, we can use the simulation results to calculate measures for buffering, reactive, and residual capacity. The MOC operates under normal conditions between times t_0 and t_{48}. At t_{48} ($t_{48} = t_d$), a disruption occurs, in this case the failure of the information fusion software. The time to execute the information fusion step in the

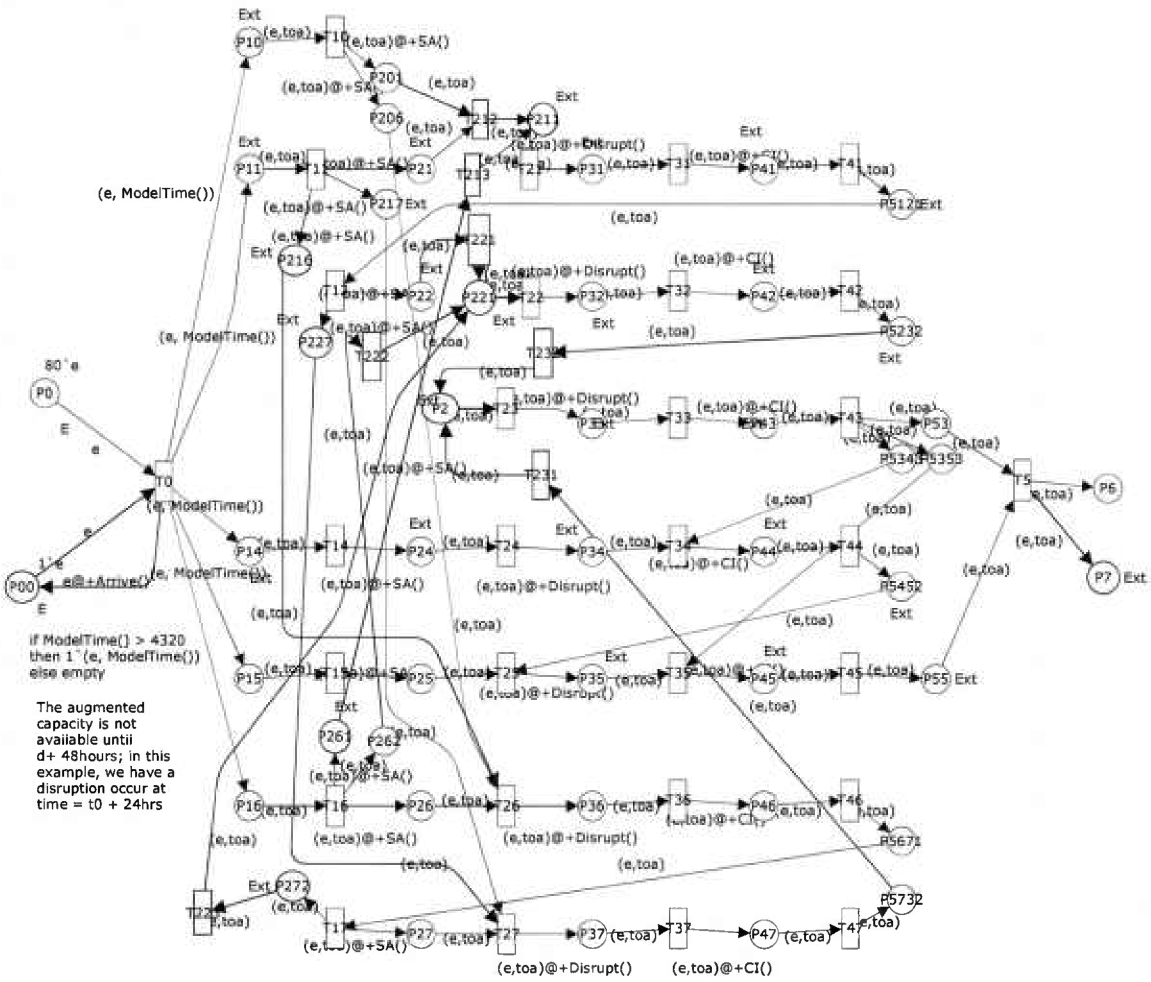


Fig. 10: Petri net for the Augmented MOC Used in Simulation

MOC Mission Order Generation capability increases as the MOC staff switches from the automated software-based approach to a manual approach. At time t_d, augmented capacity is requested, however, it takes 24 hours to stand up this augmented capacity and integrate it into the existing MOC command and control structure. Therefore, t_{rc} = 24. The augmented capacity comes in the form of an additional Intelligence Cell, and an additional Future Plans cell. Essentially, the MOC is augmenting with additional manpower to retain its capability to generate mission orders.

Figure 11 reflects the architecture's modeled simulation results during the course of the scenario. The MoP for the Generate Mission Orders MoE is shown on the vertical axis as Mission Order Generate Rate (Orders/24hrs). The time is shown on the horizontal axis. Starting at time t_0, the MOC performs under normal, pre-disruption performance levels with respect to the capability Generate Mission Orders. At this point, the MOC is capable of generating mission orders in approximately just over 4 hours. From the model results, this translates into an average of 5.67 mission orders every 24 hours. The situational awareness (information fusion) software fails at time t_{48}, and the mission order generation rate falls off dramatically as the MOC switches to manual backup procedures. At the minimum point of performance ($t_{min} = t_{53}$) , the MOC is barely at the threshold level of performance of approximately 8 hours to generate a mission order, or 3 mission orders per 24 hours. By time t_{72}, additional capacity (manpower) has been integrated to stand up an augmented future plans cell and augmented operational intelligence cell. These additional cells are able to restore a part, but not all of the original capability in terms of the mission order generation rate MOP.

In Fig. 11, the red line denotes the threshold capacity, set in this scenario as 3 mission orders generated every 24 hours, as described earlier. The green line indicates the maximum performance the MOC could achieve with respect to this capability if the augmented capacity were in place, but no disruption had occurred. This was calculated by simulating the architecture with the augmented capacity in place, but without the effects of the disruption. Comparing the Mission Order Generation Rate to the threshold value establishes the MOE for this measure.

The primary difference between the two alternative architectures under examination in this case study is that the Augmented MOC is able to generate reactive (spare) capacity, and the Base MOC is not. Maximum capacity when augmentation is available was determined by running the simulation model without the effects of the disruption, and with the spare capacity in place. This was completed by slight modifications to the inscriptions on the arcs in the Petri net shown in Fig. 10.

Using the equations for buffering, reactive and residual capacities (see Fig. 2) we find the results shown in Table 1. When operating under pre-disrupted conditions, approximately half (47%) of the MOC's capability was above the required threshold of 3 mission orders per 24 hours. During the survival phase (post disruption), the MOC was operating at the threshold of 3 orders per 24 hours. However, as the calculations indicate, no residual capacity exists, meaning that any further disruption could have resulted in catastrophic failure in terms of mission completion. The MOC was operating close to an edge in performance. Additional manpower assisted in raising MOC performance above the threshold, but did not return it to pre-disruption levels. The simulation results indicate that only restoration of the failed situational awareness software would return the MOC to pre-disruption performance levels. If the reactive capacity (the augmentation cells) were in place with no disruption, then 60% of the MOCs capacity would be above threshold.

Table 1: Determining Capacity in the MOC

Max Capacity (w/Augmentation)	7.44	V_{max}
Threshold Level	3.00	V_T
Normal Opn Level	5.67	V_2
Buffering Capacity	**47%**	
Reactive Capacity	**60%**	
Residual Capacity	**0%**	V_1

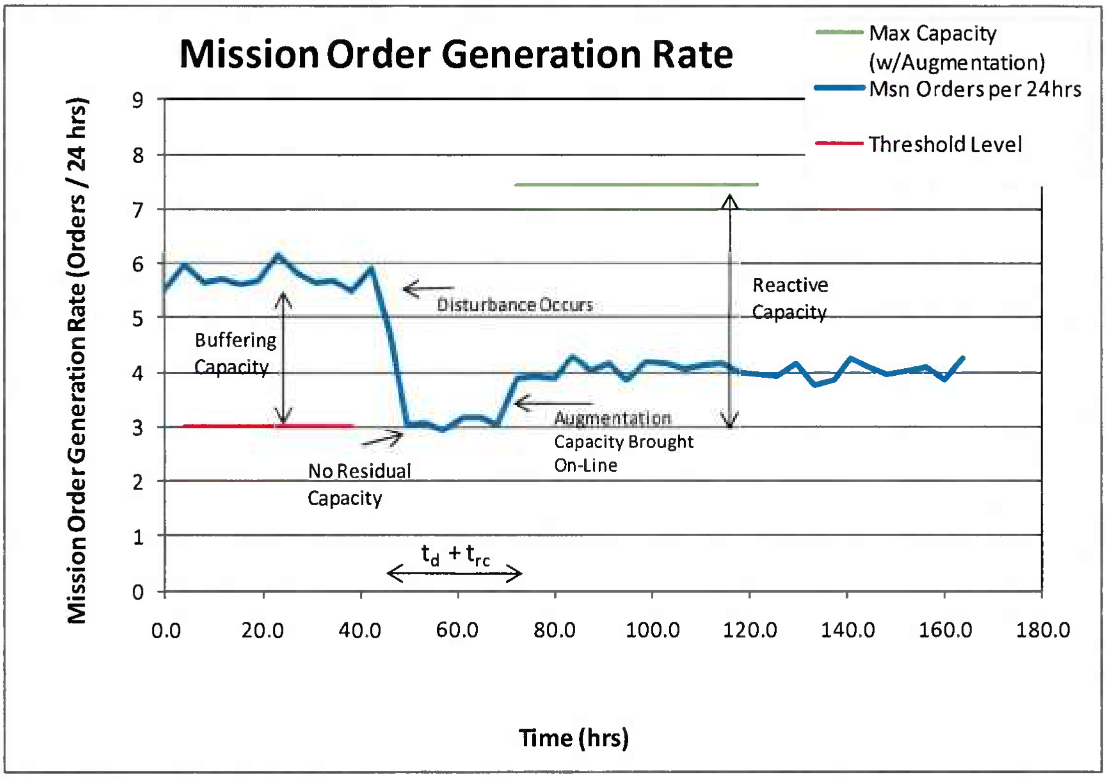


Fig. 11: Measuring Capacity in the Augmented MOC

B. *Tolerance*

As with the capacity related metrics, the rate of departure metric is determined by employing an executable model of the architecture to assess performance achieved against performance required. Recall from earlier discussion that we defined Rate of Departure (Tol_{RD})as the rate of change over time in system effectiveness in meeting its requirements after a disruption occurs. This encapsulates both the temporal aspects of resilience (t_d and t_{min}), as well as the effectiveness aspects of how the system performs with respect to its requirements and how effectiveness changes during the survival phase (post disruption).

A parameter locus is generated to account for how key parameters affecting performance may vary during the scenario. The mission order inter-arrival time is an important parameter because it represents how quickly mission orders arrive from the JFMCC. Inter-Arrival time of orders from higher HQ (JFMCC) is varied to examine the effect of queuing as the MOC executes the Mission Order process based on those JFMCC orders. The disruption involved loss of the situational awareness software supporting the information fusion stage of the MOC. Since this drives the nodes within the MOC to use manual means, the time required for the Information Fusion tasks performed is varied to reflect various manual task durations. These two variables are included in the parameter locus, shown in Fig. 12.

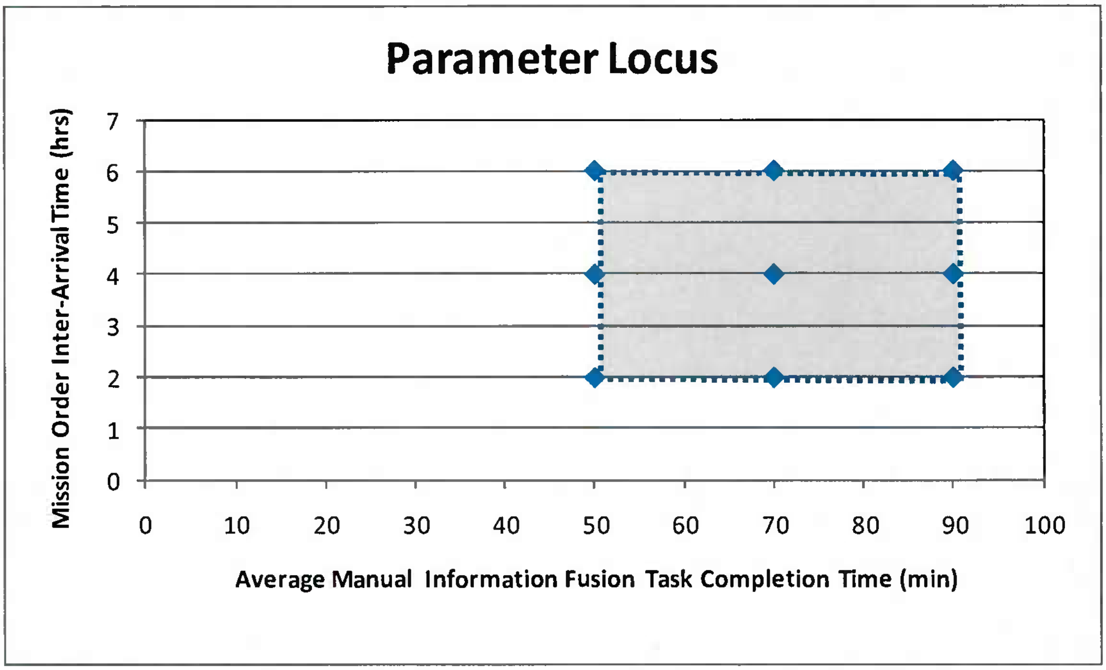


Fig. 12: Parameter Locus for the MOC

As in the targeting case study, the requirements locus is determined based on the specific variables of interest to the system under study. In the case of the MOC, the average mission order generation rate and the percent of orders delivered late to subordinate units are important. Figure 13 shows the requirements locus.

Average Mission Order Generation Rate: number of mission orders generated per 24 hours. Per the 1/3 : 2/3 planning rule, higher units do not take more than 1/3 of available time to ensure lower units can successfully execute the mission. If the MOC takes longer than approximately 8 hours to generate mission orders (3 per 24 hours), subordinate units may not be able to execute in time.

% Orders Delivered Late: Percentage of Mission Orders delivered to subordinates more than 8 hours after receipt at MOC, out of the total in the first 48 hours following the disruption. This addresses the effect on subordinate units. A threshold of 1 in 4 (25%) is established for this requirement.

Executing the architecture at each point in the parameter locus (Fig. 12) yields a locus of performance. Figure 14 displays pre-disruption performance where data is collected before the disruption occurs. In the Augmented MOC, Mission Order Generation times are well within requirements, and zero orders are delivered late to subordinate units in any portion of the parameter space.

Prior to the disruption, we can see that the Augmented MOC is very effective at the Mission Order Generation Process. Zero orders are delivered late to subordinate units within the parameter space, and the Order Generation Rate is well within the required level of effectiveness. Prior to time t_d, the performance of the system meets the requirements over 100% of the parameter space (see Fig. 16A).After the disruption occurs, the system performance meets the requirements in only 70% of the parameter space, showing a significant loss of capability after the disruption (see Fig. 16B). The MOC Decision Making architecture degraded from 100% to 70% effectiveness over a course of ~ 1 hour on average (while the event was instantaneous, the effects take time to occur fully). The rate of departure is ~33% per hour loss of effectiveness. See Fig. 16.

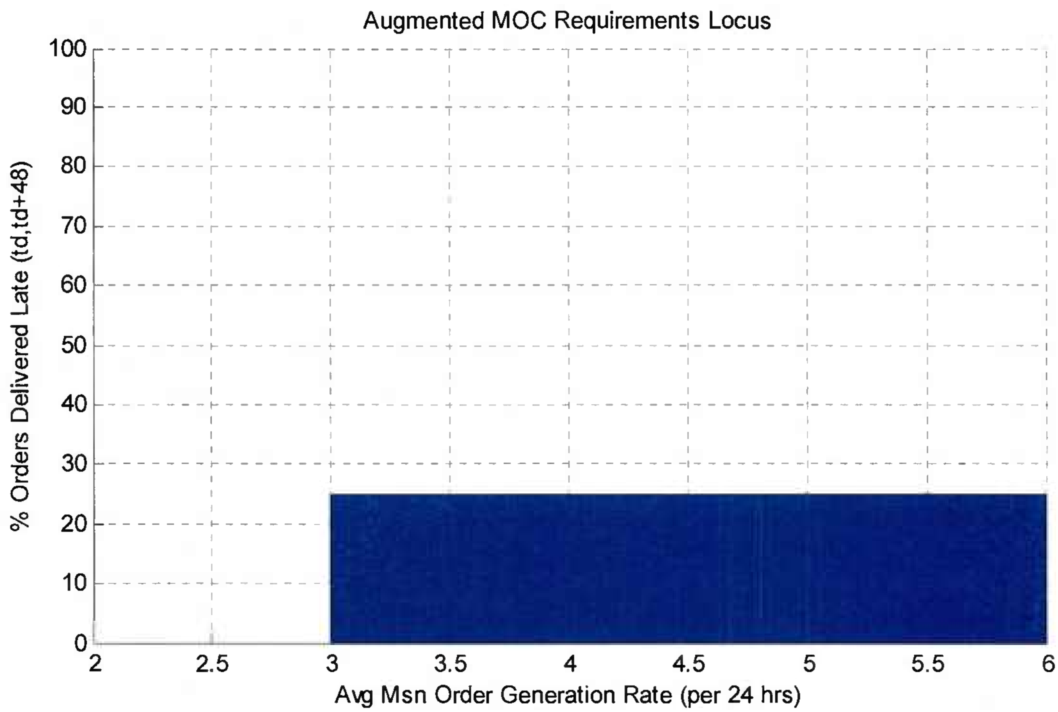


Fig. 13: Augmented MOC Requirements Locus

Executing the architecture again at each point in the parameter locus, but *after* a disruption, yields a second locus of performance. Figure 415 displays post disruption performance where data is collected during the survival phase after the disruption occurs. After the disruption, the mission order generation rate slowed as the Information Fusion process required more and more time. For certain cases, up to half of the orders in the 48 hours following the disruption were delivered late. While augmented capacity is available in the Augmented MOC, it is not available until after the augmentation cells are established, approximately 48 hours after being called for. Mission order generation is highly dependent on software to enable tasks. Loss of situational awareness software causes a reversion to manual Information Fusion methods with much longer processing times. These problems are reflected in the degraded performance seen post disruption.

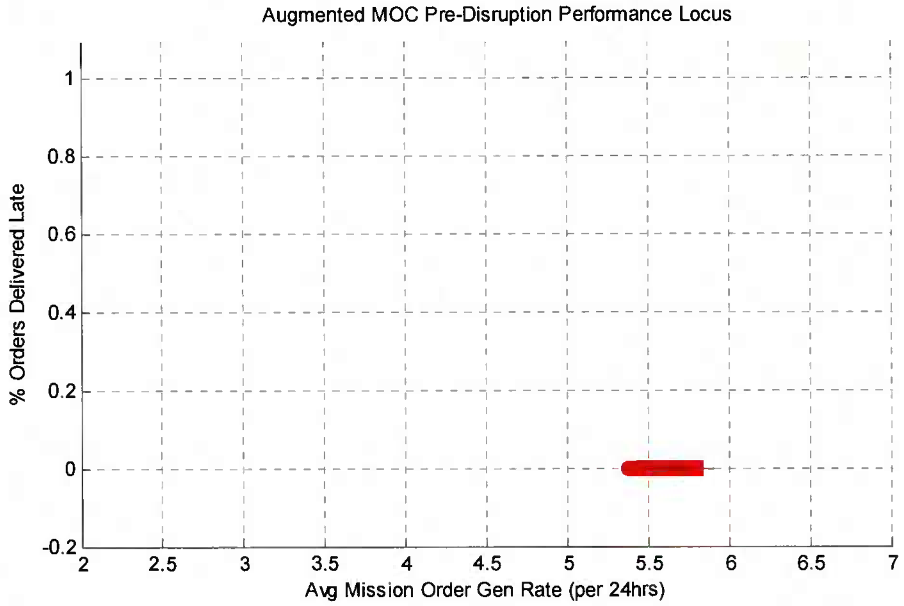


Fig. 14: Augmented MOC Pre-Disruption Performance Locus

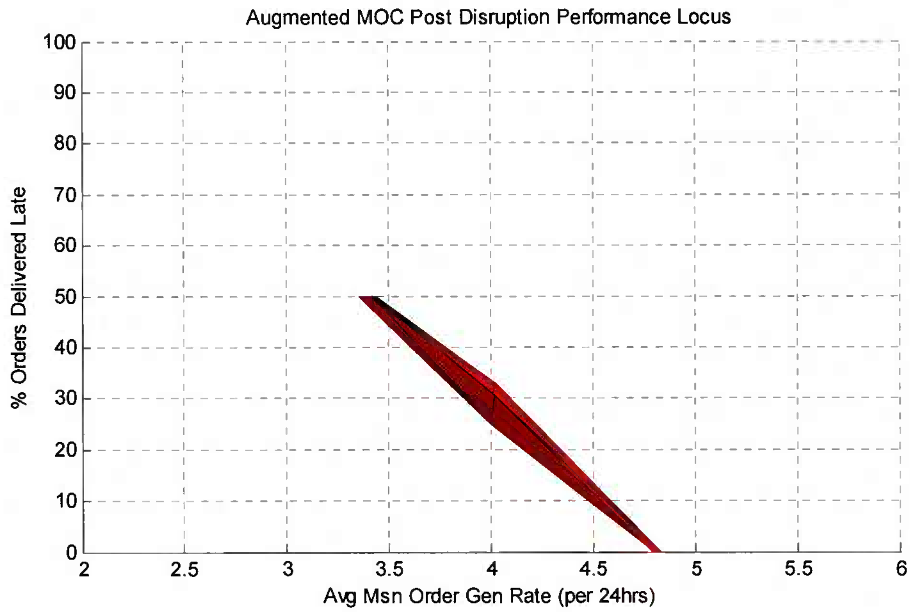
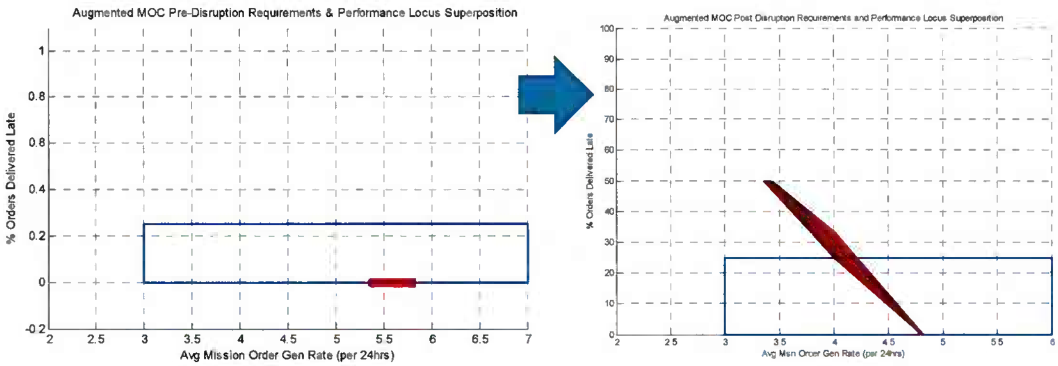


Fig. 15: Augmented MOC Post-Disruption Performance Locus

Note here that the because the augmented capacity is not available in time prior to the disruption reaching its full effect, the Rate of Departure for the Base MOC and Augmented MOC are essentially equivalent. Additionally, care should be taken in determining the time at which the minimum performance is assessed using Eq. 2 as shown in Fig. 16. Since we are typically considering stochastic systems, the absolute point of minimum performance could skew the calculation of Eq. 2. It is recommended to use the point at which the system enters this new state of degraded performance, rather than the numerically absolute minimum performance which could significantly change the denominator of Eq. 2. In the MOC case study, we used the time at which the system enters the area of minimum (i.e., disrupted) performance, versus the absolute time of minimum performance. See Fig. 17.

$$TOL_{RD} = \frac{\left[\frac{L_p \cap L_r}{L_p}, t_d\right] - \left[\frac{L_p \cap L_r}{L_p}, t_{\min}\right]}{t_{\min} - t_d} \qquad TOL_{RD} = \frac{1.0 - 0.70}{48.9 - 48} = 0.33$$



(A) (B)

Figures shown with the requirements locus as a transparent box for clarity. (The Requirements Locus is the inside of the box)

Fig. 16: Computing Rate of Departure in the MOC Case Study

In addition to being executable (supporting simulation), Petri Nets have a graph theoretic interpretation that enables the analysis of properties. The identical model used in the simulations above (see Fig. 10 - 17) was also analyzed in static form to assess other aspects of Tolerance and as well as Flexibility. As described in Section 3, examining these other aspects of Tolerance and Flexibility require an ability to determine the information flow paths which form the simple functionalities describing the overall capability under study. The information flow paths are derived from the place invariants in the architecture. CAESAR III generates the simple information flow paths associated with this net. Figure 18 shows an example simple information flow path of the Augmented MOC.

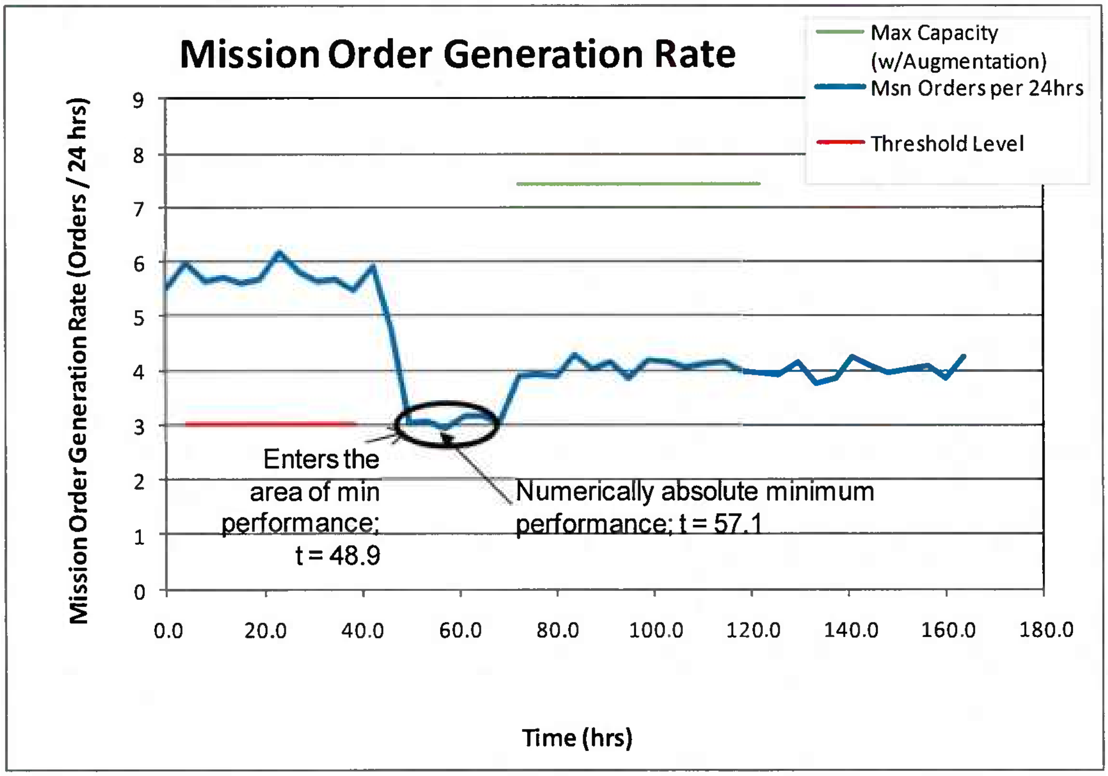


Fig. 17: Area of Minimum Performance versus Numerically Absolute Time

Fault Tolerance (Tol_{FT}) is a measure which uses the graph-theoretic properties of Petri nets. Recall that Fault Tolerance (Tol_{FT}) is the ratio of simple information flow paths which may be disrupted prior to the loss of the capability to the total number of simple information flow paths. Those elements of the sub-graph (vertices) which can be removed without disconnecting the sub-graph or eliminating the complete functionality (capability) are those that may be disrupted.

From the universal net shown in Fig. 9, there are 44 simple information flow paths, containing as many as 37 elements, or as few as 13 elements out of a total of 74 elements contained in the universal net of the Augmented MOC. This large number of information flow paths results from the high level of interconnectivity and redundancy within the augmented MOC organizational design. The Base MOC contains only eight (8) information flow paths. Table 2 shows the elements contained within each simple information flow path for both the base MOC (8 paths) and the Augmented MOC (44 paths). Note that the Augmented MOC contains all 8 of the information flow paths contained by the Base MOC.

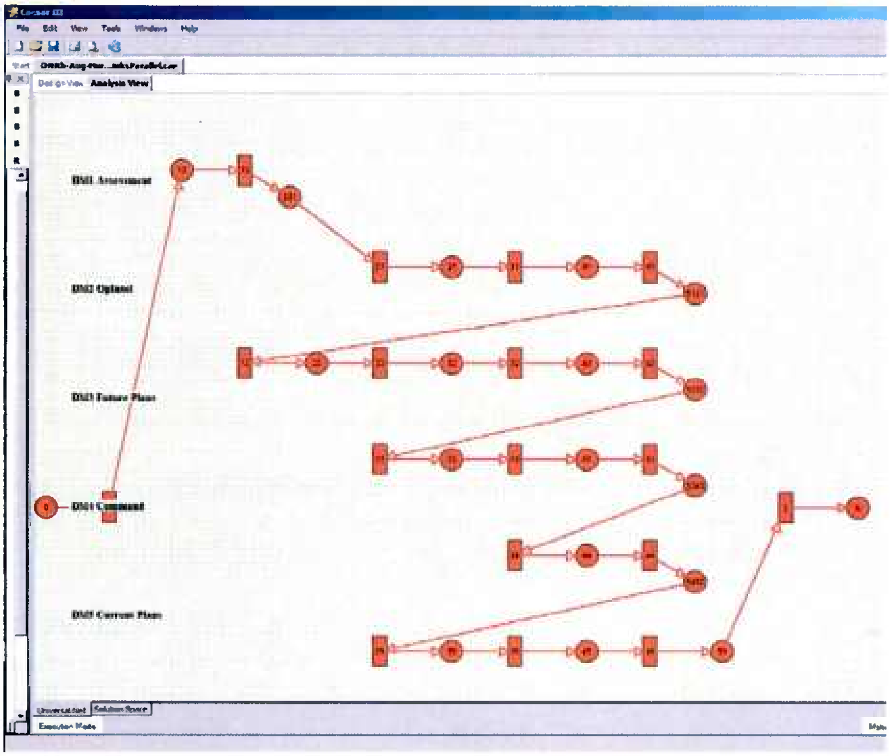


Fig. 18: Example Simple Information Flow Path of the Augmented MOC

Represented in this fashion, the methodology described in Section 2 is used to determine the cut vertices needed to compute Fault Tolerance. Any vertex that is on every path from sources S_i to sink U_j is a cut vertex. The Augmented MOC includes one source (*p0*) and two sinks (*p53, p55*). (With sinks defined as *p53* and *p55*, this eliminates the need to consider elements *t5* and *p6*, which are elements after the sinks as designated above.) We can partition the above set of information flow paths into the 14 which have *p53* as a sink, and the 30, which have *p55* as a sink. Solving for the elements common to every path from source *p0* to sink *p53*, and from source *p0* to sink *p55*, we find the following cut vertices:

sink p53 cut vertices: {p33,p43,t0,t23,t33,t43}

sink p55 cut vertices: {p45,t0,t35,t45}

There are no cut vertices common to both sinks except for *t0*. The set of non-cut vertices, (V_{nc}) includes every other element. In this example, every information flow path contains many non-cut vertices, meaning multiple flow paths exist to connect each source to its corresponding sink. In the Augmented MOC, there are 44 information flow paths so that r = 44. Since every flow path contains multiple non-cut vertices, meaning multiple flow paths exist to connect each source to its corresponding sink:

$\ell_1 \ldots \ell_{44}$each contain members of V_{nc} therefore $x_{1..44} = 1$

Table 2: Information Flow Paths in the Base and Augmented MOC

Base Path	Aug Path	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
0	0	p0	t0	p10	t10	p201	t21	p31	t31	p41	t41	p5121	t12	p22	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p53	t5	p6										
1	1	p0	t0	p10	t10	p201	t21	p31	t31	p41	t41	p5121	t12	p22	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6
2	2	p0	t0	p10	t10	p201	t21	p31	t31	p41	t41	p5121	t12	p22	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6						
	3	p0	t0	p10	t10	p201	t21	p31	t31	p41	t41	p5121	t12	p227	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p53	t5	p6										
	4	p0	t0	p10	t10	p201	t21	p31	t31	p41	t41	p5121	t12	p227	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6
	5	p0	t0	p10	t10	p201	t21	p31	t31	p41	t41	p5121	t12	p227	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6						
	6	p0	t0	p10	t10	p206	t26	p36	t36	p46	t46	p5671	t17	p27	t26	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p53	t5	p6										
	7	p0	t0	p10	t10	p206	t26	p36	t36	p46	t46	p5671	t17	p27	t26	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6
	8	p0	t0	p10	t10	p206	t26	p36	t36	p46	t46	p5671	t17	p27	t26	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6						
	9	p0	t0	p10	t10	p206	t26	p36	t36	p46	t46	p5671	t17	p272	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p53	t5	p6										
	10	p0	t0	p10	t10	p206	t26	p36	t36	p46	t46	p5671	t17	p272	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6
	11	p0	t0	p10	t10	p206	t26	p36	t36	p46	t46	p5671	t17	p272	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6						
3	12	p0	t0	p11	t11	p21	t21	p31	t31	p41	t41	p5121	t12	p22	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p53	t5	p6										
4	13	p0	t0	p11	t11	p21	t21	p31	t31	p41	t41	p5121	t12	p22	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6
5	14	p0	t0	p11	t11	p21	t21	p31	t31	p41	t41	p5121	t12	p22	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6						
	15	p0	t0	p11	t11	p21	t21	p31	t31	p41	t41	p5121	t12	p227	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p53	t5	p6										
	16	p0	t0	p11	t11	p21	t21	p31	t31	p41	t41	p5121	t12	p227	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6
	17	p0	t0	p11	t11	p21	t21	p31	t31	p41	t41	p5121	t12	p227	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6						
	18	p0	t0	p11	t11	p216	t26	p36	t36	p46	t46	p5671	t17	p27	t26	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p53	t5	p6										
	19	p0	t0	p11	t11	p216	t26	p36	t36	p46	t46	p5671	t17	p27	t26	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6
	20	p0	t0	p11	t11	p216	t26	p36	t36	p46	t46	p5671	t17	p27	t26	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6						
	21	p0	t0	p11	t11	p216	t26	p36	t36	p46	t46	p5671	t17	p272	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p53	t5	p6										
	22	p0	t0	p11	t11	p216	t26	p36	t36	p46	t46	p5671	t17	p272	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6
	23	p0	t0	p11	t11	p216	t26	p36	t36	p46	t46	p5671	t17	p272	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6						
	24	p0	t0	p11	t11	p217	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p53	t5	p6																		
	25	p0	t0	p11	t11	p217	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6								
	26	p0	t0	p11	t11	p217	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6														
6	27	p0	t0	p14	t14	p24	t24	p34	t34	p44	t44	p5454	t25	p35	t35	p45	t45	p55	t5	p6																		
7	28	p0	t0	p15	t15	p25	t25	p35	t35	p45	t45	p55	t5	p6																								
	29	p0	t0	p16	t16	p26	t26	p36	t36	p46	t46	p5671	t17	p27	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p53	t5	p6										
	30	p0	t0	p16	t16	p26	t26	p36	t36	p46	t46	p5671	t17	p27	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6
	31	p0	t0	p16	t16	p26	t26	p36	t36	p46	t46	p5671	t17	p27	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6						
	32	p0	t0	p16	t16	p26	t26	p36	t36	p46	t46	p5671	t17	p272	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p53	t5	p6										
	33	p0	t0	p16	t16	p26	t26	p36	t36	p46	t46	p5671	t17	p272	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6
	34	p0	t0	p16	t16	p26	t26	p36	t36	p46	t46	p5671	t17	p272	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6						
	35	p0	t0	p16	t16	p261	t21	p31	t31	p41	t41	p5121	t12	p22	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p53	t5	p6										
	36	p0	t0	p16	t16	p261	t21	p31	t31	p41	t41	p5121	t12	p22	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6
	37	p0	t0	p16	t16	p261	t21	p31	t31	p41	t41	p5121	t12	p22	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6						
	38	p0	t0	p16	t16	p261	t21	p31	t31	p41	t41	p5121	t12	p227	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p53	t5	p6										
	39	p0	t0	p16	t16	p261	t21	p31	t31	p41	t41	p5121	t12	p227	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6
	40	p0	t0	p16	t16	p261	t21	p31	t31	p41	t41	p5121	t12	p227	t27	p37	t37	p47	t47	p5732	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6						
	41	p0	t0	p16	t16	p262	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p53	t5	p6																		
	42	p0	t0	p16	t16	p262	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5343	t34	p44	t44	p5452	t25	p35	t35	p45	t45	p55	t5	p6								
	43	p0	t0	p16	t16	p262	t22	p32	t32	p42	t42	p5232	t23	p33	t33	p43	t43	p5353	t35	p45	t45	p55	t5	p6														

$$\mathrm{Tol}_{\mathrm{FT}} = \frac{x}{r} = \frac{\sum_{i=1}^{r} x_i}{r} = \frac{44}{44}$$

The MOC with Augmented Capacity displays maximum fault tolerance. Every information flow path can be disrupted in some way without a loss of capability. This result is not surprising, given the extensive redundancy and interconnectivity built into the Augmented MOC organizational design. Additionally, the parallel dissemination of information from *t0* fosters an embedded redundancy in the transmission of information. This parallel transmission structure is typical in military organizations, where a "warning order" is often broadly disseminated to initiate early planning activities. What this means is that multiple paths exist connecting the source to each sink, such that the elimination of a single element does not result in elimination of the overall capability.

Point of Failure Tolerance, ($\mathrm{Tol}_{\mathrm{PF}}$) examines a situation different from Fault Tolerance. $\mathrm{Tol}_{\mathrm{PF}}$ captures the relatedness of a local failure of any given element to the broader loss of functionality or loss of capability. More generally, $\mathrm{Tol}_{\mathrm{PF}}$ addresses whether an element-level failure induces a system-level failure. This is accomplished by examining the localization of failures during a disruption. Elements which are a member of only one simple information flow path are said to have localized failure effects. Table 3 associates the elements of the Petri net based architecture of the Augmented MOC with the information flow paths while Table 4 addresses the Base MOC. The equation for Tolerance yields:

For the Augmented MOC:

$$\mathrm{Tol}_{\mathrm{PF}} = \frac{\sum_{j=1}^{E} q_j}{E} = \frac{q}{E} = \frac{8}{71} = 0.11$$

For the Base MOC:

$$\mathrm{Tol}_{\mathrm{PF}} = \frac{\sum_{j=1}^{E} q_j}{E} = \frac{q}{E} = \frac{8}{50} = 0.16$$

These results imply highly non-localized failures in both the Augmented and Base MOC. Only about 11% of the elements are associated with one information flow path in the Augmented MOC, and 16% in the Base MOC. In the case of Point of Failure Tolerance, higher is better because this indicates a higher proportion of elements with localized failure effects.

Table 3: Augmented MOC: Associating Elements with Information Flow Paths

Element	# Flow Paths Associated w/Element	q_i =	Element	# Flow Paths Associated w/Element	q_i =
p0	44	0	p5232	21	0
p10	12	0	p5343	14	0
p11	15	0	p5353	14	0
p14	1	1	p5452	15	0
p15	1	1	p5671	18	0
p16	15	0	p5732	21	0
p21	6	0	t0	44	0
p22	9	0	t10	12	0
p24	1	1	t11	15	0
p25	1	1	t12	18	0
p26	6	0	t14	1	1
p27	9	0	t15	1	1
p31	18	0	t16	15	0
p32	21	0	t17	18	0
p33	42	0	t21	18	0
p34	1	1	t22	21	0
p35	16	0	t23	42	0
p36	18	0	t24	1	1
p37	21	0	t25	16	0
p41	18	0	t26	18	0
p42	21	0	t27	15	0
p43	42	0	t31	18	0
p44	15	0	t32	21	0
p45	30	0	t33	42	0
p46	18	0	t34	15	0
p47	21	0	t35	30	0
p53	14	0	t36	18	0
p55	30	0	t37	21	0
p6	44	0	t41	18	0
p201	6	0	t42	21	0
p206	6	0	t43	42	0
p216	6	0	t44	15	0
p217	3	0	t45	30	0
p227	9	0	t46	18	0
p261	6	0	t47	21	0
p262	3	0	t5	44	0
p272	9	0	E= 74	1308	q = 8
p5121	18	0			

Table 4: Base MOC: Associating Elements with Information Flow Paths

Element	# Flow Paths Associated w/Element	q =	Element	# Flow Paths Associated w/Element	q =
p0	8	0	p5121	6	0
p10	3	0	p5232	6	0
p11	3	0	p5343	2	0
p14	1	1	p5353	2	0
p15	1	1	p5452	3	0
p16	0	0	p5671	0	0
p21	3	0	p5732	0	0
p22	6	0	t0	8	0
p24	1	1	t10	3	0
p25	1	1	t11	3	0
p26	0	0	t12	6	0
p27	0	0	t14	1	1
p31	6	0	t15	1	1
p32	6	0	t16	0	0
p33	6	0	t17	0	0
p34	1	1	t21	6	0
p35	4	0	t22	6	0
p36	0	0	t23	6	0
p37	0	0	t24	1	1
p41	6	0	t25	4	0
p42	6	0	t26	0	0
p43	6	0	t27	0	0
p44	3	0	t31	6	0
p45	6	0	t32	6	0
p46	0	0	t33	6	0
p47	0	0	t34	3	0
p53	2	0	t35	6	0
p55	6	0	t36	0	0
p6	8	0	t37	0	0
p201	3	0	t41	6	0
p206	0	0	t42	6	0
p216	0	0	t43	6	0
p217	0	0	t44	3	0
p227	0	0	t45	6	0
p261	0	0	t46	0	0
p262	0	0	t47	0	0
p272	0	0	t5	8	0
			E = 50	222	q = 8

Point of Failure Tolerance is also intended to draw an architect's attention to areas in the design where greater attention may be required. In the Augmented MOC, of particular interest is that 42 of the 44, or about 95%, of the information flow paths use elements: *t23, p33, t33, p43,* and *t43*; and 30 of the 44, or almost 70%, of the information flow paths use elements: *p45* and *t45*. From Fig. 9, we can see this represents the command and current operations cells respectively. While it is natural for a military organization to rely heavily on the commander to make decisions, a disruption affecting this portion of the organizational design would have broad ranging consequences. The architect should direct attention at these portions of the architecture to determine if changes are required.

C. *Flexibility*

The final set of metrics to examine in the decision making organization case study deal with flexibility, where flexibility refers to the ability of a system to reorganize and adapt itself to changing conditions. One measure of flexibility is Cohesion, as defined by Liles [14]. Cohesion looks at the average cohesion of the individual nodes. A set of nodes with higher cohesion implies that the individual nodes are less flexible and less resilient.

We will examine flexibility where each decision making organization within the MOC identified in Figs. 6 and 7 is treated as a node (i.e. Assessment, Operational Intelligence, Future Plans, Command, Current Plans, and Current Operations). Executing Eq. 5 and Eq. 6 yields the results shown in Fig. 19. These results show that the Augmented MOC is less cohesive than the Base MOC and therefore more flexible.

Eq (5) $$Coh(n_{ki}) = \frac{z_{ki}}{x_{ki}}$$

Eq (6) $$Coh(f_k) = \frac{\sum_{i=1}^{m} Coh(n_{ki})}{m}$$

Cohesion (Mult Nodes) Augmented MOC

Node	Inputs	Outputs	Paths	Coh (nki)
DM1	1	2	2	1.00
DM2	3	3	5	0.56
DM3	3	2	4	0.67
DM4	2	3	6	1.00
DM5	2	1	2	1.00
DM6	3	1	3	1.00
DM7	3	3	5	0.56
DM8	3	2	4	0.67

m = 8 $Coh(f_k)$ = 0.81

Cohesion (Mult Nodes) Base MOC (non- Augmented)

Node	Inputs	Outputs	Paths	Coh (nki)
DM1	1	1	1	1.00
DM2	2	1	2	1.00
DM3	1	1	1	1.00
DM4	1	3	3	1.00
DM5	2	1	2	1.00
DM6	3	1	3	1.00

m = 6 $Coh(f_k)$ = 1.00

Fig. 19: Calculating Cohesion in the MOC

These results are somewhat intuitive. In this case study, we are essentially adding capacity for the intelligence and future planning functionality through augmentation cells which provide a redundant capability in those areas. This should naturally increase the flexibility of the MOC as an organization. This approach quantifies that increase.

Liles [14] also introduces a second measure of flexibility, which we have renamed as Common Use. Recall that CU measures the extent of reuse of the elements to support multiple simple functionalities that comprise the overall capability. Tables 3 and 4 associate the number of simple functionalities that each element is a member. Executing the equation for the Common Use yields the following:

Augmented MOC:

$$\text{Common Use (CU)} = \frac{\sum_{j=1}^{E} A}{E} = \frac{1308}{74} = 17.7$$

Base MOC:

$$\text{Common Use (CU)} = \frac{\sum_{j=1}^{E} A}{E} = \frac{222}{50} = 4.4$$

From Common Use alone, it is difficult to determine whether 4.4 vs. 17.7 is an improvement or not. This is because there are 44 information flow paths in the Augmented MOC, but only 8 in the Base MOC. Therefore, the numbers for Common Use will be inherently different. The next section helps explain these metrics in a more comparable fashion to support evaluation.

We defined Proportion of Use as the relative proportion of elements used by any given simple functionality to deliver the overall capability. We note two principal advantages to this metric. First, it describes what proportion of the elements is contained within the average simple functionality of a capability. For example, does the average functionality use a relatively small (say 10%) or a relatively large (say 80%) of the elements? As proportion of use increases, a disruption to a given element within a capability is more likely to have broad ranging effects. Systems with high proportion of use are more difficult to reorganize (less flexible), because each element is more related to each functionality. Second, proportion of use normalizes the Common Use such that one can compare different architectures from a common perspective. This allows us to determine whether a particular value for Common Use is comparatively high or low. Figures 20 and 21 show the results of computing Proportion of Use for the Augmented and Base MOC alternatives.

For the MOC with Augmentation, Proportion of Use is 0.4, meaning that each simple functionality involves about 40% of the elements required to deliver the capability. In the base MOC without augmentation, each simple functionality involves approximately 56% of the total elements. From this perspective, we can say that the augmented MOC is more flexible. In the augmented MOC, a disruption to a given element can be expected to affect about 40% of the overall functionality of the system under evaluation. In the base MOC, a disruption to a given element can be expected to affect about 56%.

Inf Flow Path	# Elements Contained by ℓ_i	Inf Flow Path	# Elements Contained by ℓ_i
ℓ = 1	27	ℓ = 24	30
ℓ = 2	27	ℓ = 25	37
ℓ = 3	37	ℓ = 26	31
ℓ = 4	31	ℓ = 27	37
ℓ = 5	37	ℓ = 28	31
ℓ = 6	31	ℓ = 29	36
ℓ = 7	19	ℓ = 30	30
ℓ = 8	13	ℓ = 31	37
ℓ = 9	27	ℓ = 32	31
ℓ = 10	26	ℓ = 33	29
ℓ = 11	27	ℓ = 34	23
ℓ = 12	27	ℓ = 35	37
ℓ = 13	26	ℓ = 36	31
ℓ = 14	27	ℓ = 37	37
ℓ = 15	19	ℓ = 38	31
ℓ = 16	27	ℓ = 39	37
ℓ = 17	27	ℓ = 40	31
ℓ = 18	27	ℓ = 41	37
ℓ = 19	27	ℓ = 42	31
ℓ = 20	19	ℓ = 43	29
ℓ = 21	37	ℓ = 44	23
ℓ = 22	31	Σ B = 1308	
ℓ = 23	36	E = 74	
		r = 44	

$$\text{PoU} = \frac{\dfrac{\sum_{i=1}^{r} B_i}{E}}{r} = \frac{\sum_{i=1}^{r} B_i}{rE}$$

$$= \frac{1308}{74*44} = \frac{1308}{3256} = 40.4\%$$

Fig. 20: Proportion of Use in the Augmented MOC

Inf Flow Path	# Elements Contained by ℓ_i
ℓ = 1	27
ℓ = 2	27
ℓ = 3	37
ℓ = 4	31
ℓ = 5	37
ℓ = 6	31
ℓ = 7	19
ℓ = 8	13

Σ B = 222
E = 50
r = 8

$$\text{PoU} = \frac{\dfrac{\sum_{i=1}^{r} B_i}{E}}{r} = \frac{\sum_{i=1}^{r} B_i}{rE}$$

$$= \frac{222}{50*8} = \frac{222}{400} = 55.5\%$$

Fig. 21: Proportion of Use in the Base MOC

MOC Case Study Resilience Results

In this case study, we have applied the individual components of the resilience evaluation approach for both the Base MOC, and the Augmented MOC alternatives. The MOC is designed as a series of Decision Making Nodes, with each node as a five stage decision making structure with interactions between nodes. This architecture was transformed into an ordinary Petri Net using the theory outlined in Remy et al. [12]. Necessary logic and instrumentation were added to the Petri Net such that it became an executable form of the MOC architecture suitable for behavioral and performance analyses.

The capability under study was the capability to Generate Mission Orders, where the threshold performance level was to generate the order within 8 hours of receipt from the High Joint Command.

It is critical to consider the resilience 'of what' 'to what,' focusing on the resilience of a capability to a disruption in a particular environment (under what conditions). In this case study, we examined the resilience of the MOC's capability to 'Generate Mission Orders' to the disruption 'loss of situational awareness software.' When a new release was received, the update caused both versions to crash, and attempts to restart were unsuccessful. The MOC, transitioned from automated procedures based on the software, to manual procedures, and called upon augmented capabilities in the form of additional Operations Intelligence and Future Plans cells, which required additional time hours to establish. However, this augmented capability could not be established until well after the disruption had induced its full effect.

For both cases of the MOC, the disruption brought the MOC's capability to the brink of not meeting the threshold. If another disruption occurred before the augmented capacity could be brought online, the MOC would have been incapable of completing one of its key capabilities, the generation of mission orders. The Augmented capability did return a portion of the MOC's mission order generation capability, but not back to pre-disruption levels. Table 5 reports the results for each metric in the base MOC and the Augmented MOC.

Table 5: Resilience Metrics for the Base and Augmented MOC

Attribute	Metric	Measures	Question Answered	Augmented MOC	Base MOC
Capacity: *"the ebility to operate et e certain level es defined by a given meesure."*	Buffering Capacity	Available capability margin between current operating levels and e defined minimum threshold operating level et the time preceding a disruption.	Can a disruption be absorbed with immediately evailable (on-hend) resources?	47%	47%
	Reactive Capacity	Available capability margin between maximum operating levels (i.e. including any spare cepecity) and a defined minimum threshold operating level.	Can a disruption be ebsorbed with the addition of spare capacity?	60%	0%
	Residuel Capecity	Available capability mergin between operating levels et the end of the survival phese end e defined minimum threshold operating level.	Given survival, how vulnerable is the system to e follow-on disruption thet occurs before the system can recover?	~0%	~0%
Tolerance: *"the ebility to degrade gracefully efter e disruption"*	Rate of Departure	Rate of change in system performance with respect to its requirements (ie rate of loss of effectiveness) after a disruption.	What level of capebility is lost per unit of time during the survivel phase?	0.33	0.33
	Fault Tolerance	The ratio of simple functionalites which may be disrupted without a loss of capebility to the total number of simple functionalities.	How many simple functionalities can be disrupted prior to losing the cepebility. Primarily e tool to draw erchitects ettention to key areas in the design.	1.0	1.0
	Point of Failure Tolerance	Relatedness of failures at the element level to an overall loss of capability	Are element level failures relatively localized, or do failures incur broed system-level effects? Primerily e tool to draw the architect's attention to key design areas.	0.11	0.32
Flexibility: *"the ebility of a system to reorganize its elements to maintein its capabilities"*	Cohesion	Relatedness of the elements within e node or module which support a given capability	How difficult is it to reorganize the system at the node / module level?	0.81	1
	Common Use	Extent of common use of the elements among the simple functionalities which support the overall capability.	Can a system execute multiple functionalites concurrently, or is it limited by competition for resources?	17.6	4.4
	Proportion of Use	The ratio of the total elements used by eny given simple functionality to deliver the overall capability	Are most of the elements needed for a given functionality, making it more difficult to reorganize?	0.40	0.56

The architect along with the overall development team must consider which aspects of resilience are most important to the system under consideration. Table 10 assists the architect to determine which aspects of resilience are most applicable to the architecture definition and resilience issues at hand. In the case of the MOC and capacity, the most appropriate metric is Buffering Capacity. As was shown, reactive capacity is not available in time to play a role in the survival phase, although it does distinguish the two candidate architectures and, once in place, the augmented capacity offers a number of benefits. Further, since this was a non-malicious type of disruption, the residual capacity is of less concern because a follow-on disruption is not necessarily likely. For Tolerance, the most appropriate metric is again Rate of Departure. In this case, Rate of Departure directly measured the time sensitive nature of the MOC's capability of generate mission orders by assessing the rate of generation and the number of "late" orders delivered to subordinate units. In terms of flexibility, Proportion of Use was also selected because it directly addresses the ability of the system to be reorganized based on the average use of the elements across the simply functionalities supporting that capability. The cohesion metric is not as useful, because the disruption is likely to take effect much more quickly than any reorganization could occur. This makes cohesion a metric potentially more useful in the 'recovery' phase of resiliency. Additionally, the overlap in the Common Use and Proportion of Use was discussed, leading to a selection of Proportion of Use for this assessment.

Having determined which metrics are most important, resilience can be considered from an intersecting requirements and performance locus perspective. The two alternative architectures can be compared in this manner. Determining the resilience requirements locus requires value judgment. The example is shown with a requirements locus:

33% Buffering Capacity

< 50% Rate of Departure (Tolerance)

< 50% Proportion of Use (Flexibility)

Figure 22 shows the results of the overall evaluation. The Augmented MOC meets the resilience attribute requirements of capacity, tolerance and flexibility making it the preferred candidate architecture from the point of view of resiliency. The Base MOC lacks required flexibility but meets the other requirements.

Figure 23 shows a comparison that better captures the difference of augmentation between the two candidate architectures. The Augmented MOC is able to bring reactive capacity on-line, whereas the Base MOC is not. Using the same perspective, the following graph shows the impact of considering capacity in terms of reactive capacity rather than buffering capacity. While these results do not change the overall resilience comparison of alternative architectures, it is shown here as another possible viewpoint since the two alternatives differ in terms of reactive capacity.

As we consider the performance of the two designs, Base MOC, and Augmented MOC, the evaluation framework allows us to make useful, quantitative comparisons. In the case of capacity, the two designs have equivalent buffering capacity and residual capacity. While the augmented MOC has greater reactive capacity, the augmented MOC cannot bring that reactive capacity on line fast enough to make a difference in the survival phase. In the case of point of failure tolerance, the Base MOC actually performs better. This is because its failures are the most localized. More specifically, about 1/3 (0.32) of the base MOC element failures are localized, as compared to ~ 1/10 (0.11) of the augmented MOC. The greater interconnectivity of the augmented MOC accounts for this difference. In the case of flexibility, the augmented MOC performs best in terms of Proportion of Use. A smaller proportion of its elements, on average, are needed for a given functionality, as compared to the base MOC (40% vs. 56%).

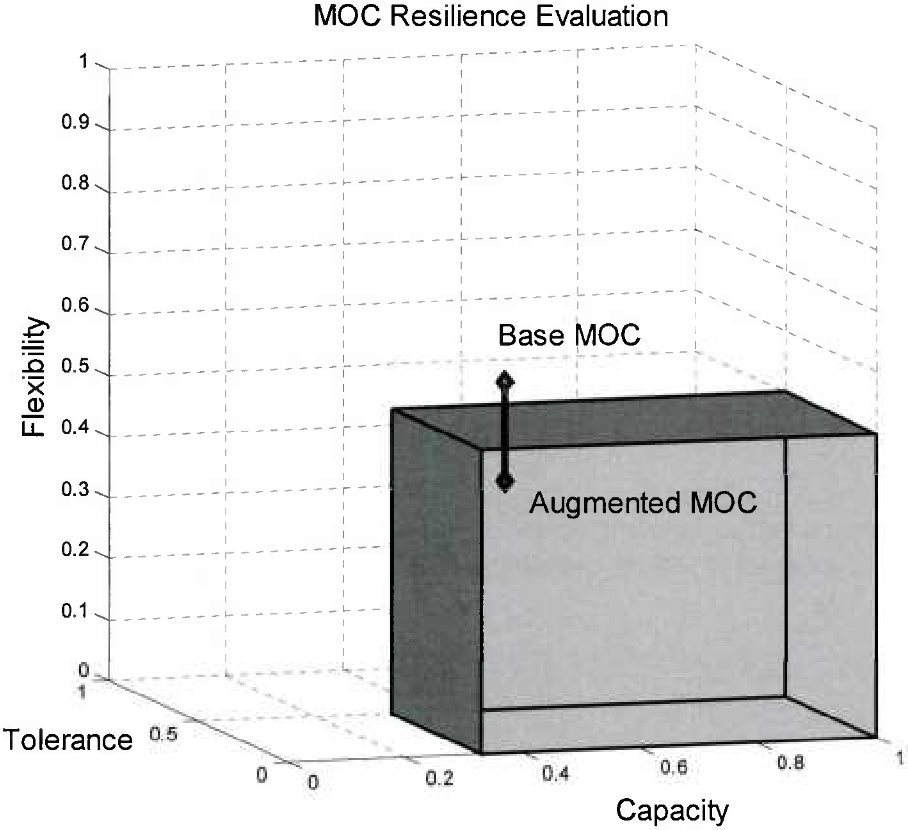


Fig. 22: Resilience Evaluation of the Base and Augmented MOC

The primary advantage of the augmented MOC (additional reactive capacity) is not relevant during the survival phase; the augmented and base MOC perform equivalently in terms of buffering capacity. However, the reactive capacity does allow the augmented MOC to restore performance above the (T), making a future disruption less likely to have catastrophic effect when compared to the base MOC. In terms of tolerance, the augmented MOC performs worse because its failures are less localized and therefore more likely to have broader, system level effects. In terms of flexibility, the augmented MOC performs better; the fact that fewer elements are needed for the average functionality will make the augmented MOC more easily reorganized.

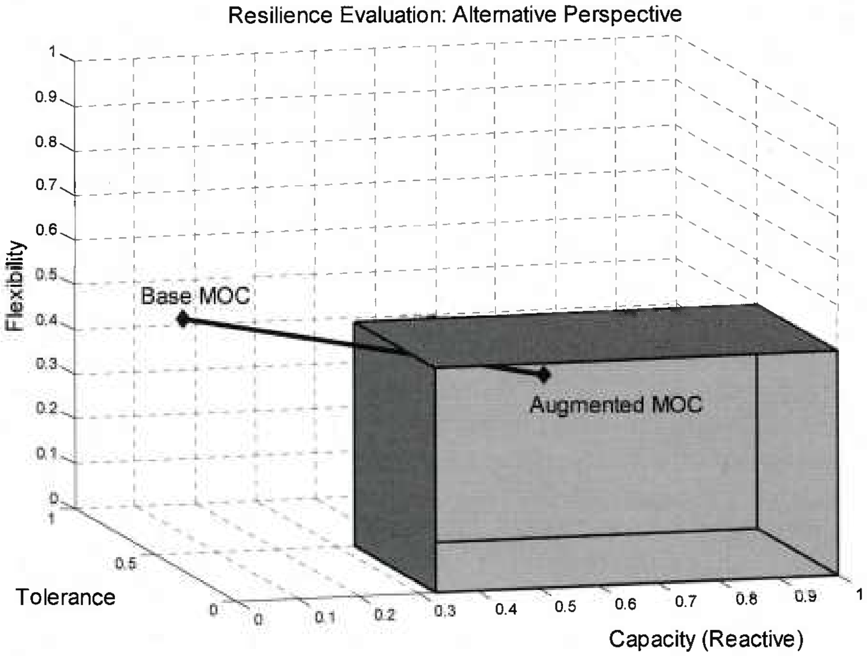


Fig. 23: An Alternative Resilience Evaluation for the Base and Augmented MOC

This case study provided a simple demonstration of how to compare alternative architectures in terms of resilience. It applied the same methodologies as in the targeting case study. A MOC case study was developed in a 5 stage architecture model for representing Decision Making organizations approach as supported by CAESAR III. CAESAR III supports the translation of the 5-stage DM model to Petri Net form. From this point forward the methodology is identical to that used in the targeting architecture case study. Resilience was evaluated using the architecture by: identifying the appropriate measure for each resilience attribute and mapping the architecture performance of each measure against an overlaid requirements locus. The resilience-related potential improvements to the design were highlighted and alternatives can now be quantified or compared. This case study demonstrated that the approach to evaluating resilience is robust to differing architecture methodologies.

2.5 Comments and Conclusions

This research has produced a quantitative means to evaluating the expected resilience of a organizational architecture performing command and control. The key idea has been to use measures for each of the attributes of resilience, and then to combine these measures into a holistic assessment. By representing the architecture in a rigorous way using Petri nets, the approach supports simulation of the architecture and the analysis of properties based on structure. This allows us to examine the expected performance (by executing the Petri net based architecture) and structural characteristics, such as analysis of the simple information flow paths of the architecture. Two case studies demonstrating the approach are available in [6]. However, this work focused on the survival phase of resilience and did not address recovery or avoidance phases. While flexibility does in part address characteristics beneficial during a recovery phase, such as the ability to reorganize, further research is needed to identify a complete end-to-end assessment of resilience that would include the avoidance, survival, and recovery phases.

III. INTEGRATED COMMAND AND CONTROL PLANNING

3.1. Introduction

Military commanders always seek to maximize the effects of their organization's components by properly arranging them in time and space to achieve integration. The speed and complexity of modern warfare have only magnified the difficulty in achieving integration [15], [16]. This challenge is well documented and variously referred to as the need for: synchronization, synergy, unified action, coordination, and/or collaboration in military planning and military command and control (C2) in general. Many recent military policy and strategy documents make reference to the necessity of integration and related concept as a method to mitigate rising complexity and the challenge of diverse mission requirements [17], [18], [19]. Reports and critiques of shortcomings in modern military operations also point to integration as a concern that has yet to be fully addressed [20], [21]. A great deal of research and development emphasis has been placed on integration. These efforts have focused on increasing information sharing and enabling knowledge sharing between organization components [22], [23], [24]. Even as knowledge sharing barriers diminish, the challenge of efficiently building common knowledge in time constrained military planning remains [25]. New approaches to military planning and the supporting command and control architectures will be necessary to maximize the benefit provided by new capabilities of knowledge sharing.

The objective of this paper is to describe an approach to military planning which will increase integration between cooperating organizational components, which are termed domains, and will result in better integrated courses of action (COAs). The approach involves investment of additional time early in the planning process to develop a common conceptual model of the operational environment between domains. This approach is contrasted with traditional approaches of separate domain COA development and subsequent de-confliction (iterative adjustment of domain COAs to remove activities that have severe negative impact on the other domains' effectiveness). To demonstrate the feasibility of the proposed approach, a modeling methodology was developed which relates the modified planning process to the performance of the resulting developed COAs. Section 3.2 describes the concepts of conceptual models, planning, and design, as they related to this effort. Section 3.3 introduces the new approach for increasing integration through common conceptual model building. Section 3.4 illustrates the modeling methodology that is used to demonstrate the feasibility of the proposed approach. Section 3.5 presents the results, while Section 3.6 summarizes the work and suggests areas for continued research.

3.2 Conceptual Models, Planning, and Design

In order to explore inter-organizational integration, several cooperating domains were considered. These domains are separate functional components of an organization or coalition cooperating towards a common goal(s). Complete integration of the domains' COAs is then defined as follows: ***COAs in which all participating entities act as one organization in pursuit***

of common goal(s); A set of COAs in which no higher performance can be obtained by changing the actions taken and action timing in any involved domain COA. During military planning, the domains are in the process of creating and evaluating COAs. For COA integration, how and when to share information must be considered.

A great deal of research has been done on information sharing between organizations. This research area is extremely broad, potentially covering the fields of management, organizational communication, knowledge management (KM), information technology, and others. One theme which is common to a majority of research in these fields is the delineation of data/information and knowledge [26], [27], [28]. Related to this is the idea of an individual's or organization's conceptualization of the situation at hand, or the operational environment. Data and information are used to produce organizational knowledge of a situation. Through organizational processes, this knowledge is used to create a conceptual model of the operational environment for which military planning is taking place. [28] This relationship is shown in Figure 24. Sharing of data/information is a requirement before sharing of knowledge can be considered. Likewise, knowledge sharing is necessary but not sufficient for conceptual model sharing. The generic term "elements" is used for information/data, knowledge, and conceptual models components.

During military planning each domain is building a conceptual model of the operational environment. Organization information, knowledge, and conceptual models are evolving during the planning process until decisions are made by the commander to approve specific aspects at certain points. Based on this understanding of how the operational environment works, each domain will choose a COA which best meets the commander's and/or higher authorities' specified criteria.

Fig. 24: Organization Information, Knowledge, and Conceptual Models

There are two primary considerations in developing processes to increase inter-domain COA integration: what is shared, i.e., conceptual models, knowledge, or information, and when in the process this sharing is attempted, as shown in Figure 25. The choice of when in the process to share elements affects whether or not the specific element has been approved by the domain commander. In addition, for conceptual models and knowledge, there is the choice of whether or not and when to attempt inter-domain agreement on a specific element.

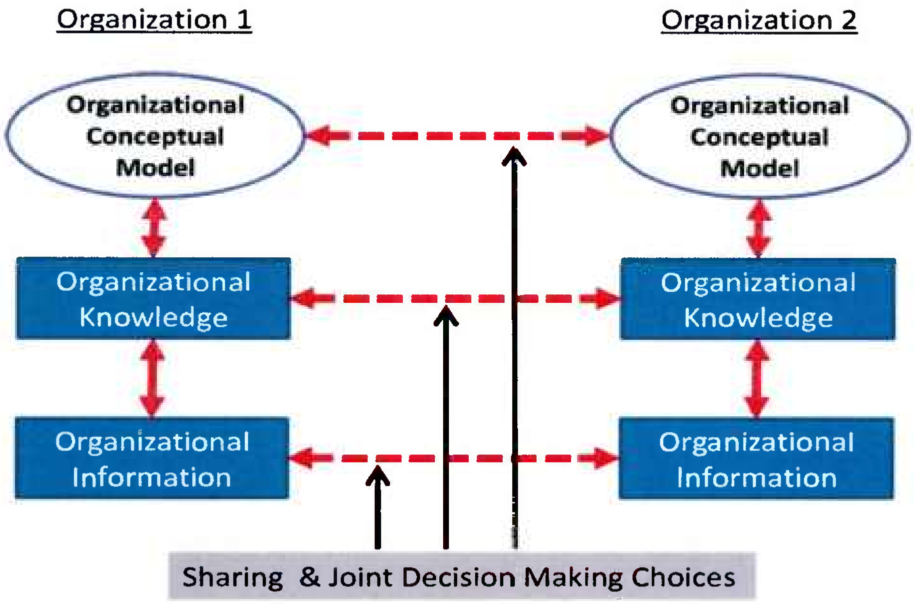


Fig. 25: Element Sharing and Joint Decision Options

Current military planning approaches were explored to understand the potential points for element sharing. During military planning, two complementary activities proceed concurrently, planning and design. Design (also called operational design) involves problem setting and framing and is normally associated with the commander. Organization staff usually conduct planning which is a procedural problem solving process. The United States doctrinal planning processes differ slightly between levels of war and organization/service but are generally similar; examples include the Joint Operations Planning and Execution System (JOPES) and the Military Decision Making Process (MDMP). Most NATO and Western military organizations use similar prescriptive process models. There have been suggestions to include more aspects of Naturalistic Decision Making theory in military planning processes, which would more explicitly integrate planning and design [29], [30], [31]. These suggestion are still under debate and there have been few significant changes to the process since World War II [32]. United States military descriptions of design focus not on procedure steps but on the results of design which frame the staff planning effort. The design framework in United States military joint planning documents is Center-of- Gravity (COG) analysis [19]. Alternative frameworks include: Effects-Based Operations [33], [34], [35], Operational Net Assessment [36], and Systemic Operational Design [37], [38], [39].

In many military planning situations time is a critical factor. In time sensitive planning situations, a trade-off must always be considered between planning time and plan quality/integration. This is summarized well in the United States Army's new field manual on operations: *"Taking more time to plan often results in greater synchronization; however, any delay in execution risks yielding the initiative— with more time to prepare and act—to the enemy."* [40] In planning situations where time is less important, time inefficient processes of inter-domain adjustment can be used. In the more rigorous time constrained environment, full inter-domain de-confliction may not be possible within the time allowed for planning. For a new approach to be considered for use in time sensitive planning, it must not significantly increase the required time for planning. Whether explicit or not, the processes of planning and design are creating an organizational conceptual model of the operational environment. In current military doctrine, this occurs mainly in the first stage of the planning process, Mission Analysis, and the concurrent design activities. The organization conceptual models may be modified during COA development, comparison, and analysis but formulation of the model has largely already occurred. As each domain creates a unique conceptual model of the environment, the stage is set for difficulty in resolving conflicts between domain courses of action later in the process. During conflict resolution, also called de-confliction, domains will try to resolve selected actions which cause negative effects on other domains. The understanding of cross-domain effects will be based on the differing organizational conceptual models making mutual adjustment difficult. United States military planning doctrine does not explicitly define a methodology for inter-domain planning integration [40], [41], [42]. The importance of planning integration is articulated but no specific approach is suggested.[1] The traditional method for producing an integrated COA is to develop and approve domain COAs and then begin the time consuming process of mutual adjustment coordination[2] to obtain the best performing (criteria determined by the commander) integrated COA. Domains do share information during the planning process but the usefulness can be limited because of information ageing and concurrency issues. This process clearly breaks down in a time constrained environment where the integration level of the COAs is ultimately determined by the time available for mutual adjustment coordination. This is the reality of current United States military planning processes shown in Fig. 26. The process block entitled "Informal design coordination" represents the process of coming to some level of common agreement on a conceptual model of the operational environment. This must take place to have a meaningful dialog on COA changes that increase overall inter-domain effectiveness.

Attempts to solve the integration challenge in military planning have been largely focused on increasing information sharing and enabling knowledge sharing. Significant resources have been applied to increase the number and interoperability of information systems to allow greater information flow between domains [44], [45], [46]. Many efforts are underway to enable and streamline knowledge sharing through creation of common ontologies and related capabilities [47], [48], [49]. Other efforts have focused on enforcing joint conceptual frameworks through use of common decision support systems among domains [50], [51]. Another approach has been to reduce the partitions between domains [52]. These various approaches will contribute to an eventual solution but continued emphasis indicates that challenges remain. Once the capability to share knowledge efficiently has been realized, there will still be the requirement for inter-organizational processes (when and what knowledge to share) to encourage integration.

[1] United States military planning doctrine does not ignore potential inter-domain interactions during COA development but there is no formal method for identifying inter-domain effects.

[2] Mutual adjustment coordination is the most resource intensive of the standard coordination methods described by Thompson [43].

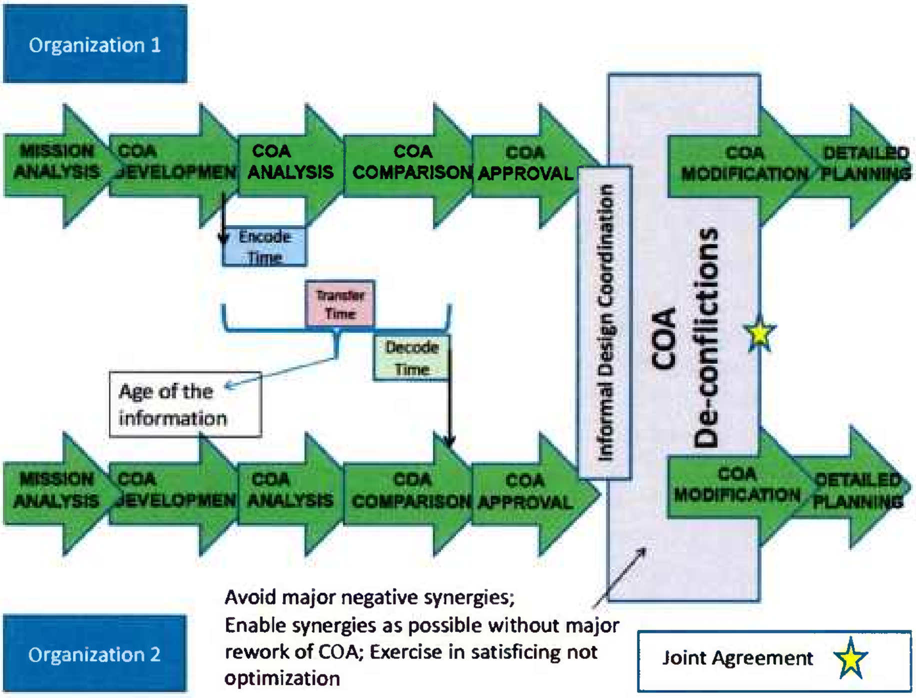


Fig. 26: Current Military De-confliction Approach

If we consider military planning in a generic sense, it is a problem solving and design process. Inter-organization military planning is then related to group problem solving, cooperative work, and concurrent/distributed design processes. Research in these non-military fields then provides some insight into approaches for integration. Emerging research in these areas indicates there is a connection between agreeing on a common conceptual model and the integration level of the resulting product [53], [54], [55]. In their research COA Action, Klein et al. demonstrated that for experts in tactical decision making having a common conceptual model enables joint option awareness [56]. Joint option awareness is the understanding of how well a COA meets the commander's criteria and the underlying aspects which affect how well a COA meets the criteria. In the Collaboration Evaluation Framework (CEF) research, Aldeman et al. [57] demonstrate through experimentation Thompson's concept of collaboration methods becoming more resource intensive as they progress from standardized to planned to mutual adjustment. In experiments with tactical level military planning scenarios, it was shown that changing collaboration tasks from mutual adjustment to planned or standardized coordination methods lowered the communication and cognitive resource costs [57]. This would indicate that building a common conceptual model lowers the resource cost of integrating COAs.

3.3 An Approach to Integrated Planning.

Separate domain conceptual models make integration very difficult and a common model increases integration [58]; therefore the goal is clear: a process that will facilitate common conceptual model creation during military planning without significantly increasing the time required. The proposed approach is based on creating a common conceptual model of the operational environment among all domains prior to developing COAs. Important to the overall concept is the acknowledgement that the domains seek to establish a common conceptual model. Although information and knowledge sharing is required, this is the means and not the end. Current approaches toward integration are based on increasing knowledge sharing: Commanders are sharing knowledge with other commanders, Commanders are communicating knowledge to their staff, and Staffs are sharing knowledge with other staffs. The exchange of knowledge implicitly and slowly adjusts domain conceptual models, but COAs that are initially based on domain conceptual models and then de-conflicted create the burden of changing domain conceptual models after they have been formed. In contrast, the proposed approach is based on integrating the necessary components of domain conceptual models before beginning to develop courses of action.

The proposed approach is centered on consensus building between domains during the operational design process and related planning activities. This approach is therefore termed "Co-design" as it describes a cooperative operational design process among domain participants. Five stages were developed to build incrementally the common conceptual model during mission analysis. This allows domains to agree on essential conceptual model elements one increment at a time to simplify consensus building. The five stages and the conceptual model component delineation were chosen to align with existing concepts in operational design. The five steps, termed "design coordinations", are: 1. Objective(s) and metric(s), 2. Key Influencers of objective(s), 3. Adversary and environment potential actions, 4. Organizations' (Domains') potential actions, and 5. System structure (interactions, constraints, synergies). These five steps are envisioned as enabling joint conceptual model creation. To these, three more design coordinations are added to facilitate the overall integrated COA development process: Step 0. Agreement on Coordination Approaches (if not specified by previous agreement), Step 6. Develop Integrated COA Actions, and Step 7. Establish COA Action Timings. The entire process between two domains is shown in Figure 27. Higher headquarters guidance and its potential effect on any point in the process are explicitly shown.

An attempt was made to lower the potential implementation burden of the new approach through use of existing planning and design processes as much as possible. First the necessary components of a common conceptual model to allow integrated COA creation were identified. These components were then related to the conceptual model components which are commonly created by commanders during operational design. In turn, the necessary inputs for each component of the commanders' design from standard military planning process activities were determined. An example of this information/knowledge relationship is shown in Table 6 for step 2 of Design Coordination. This example specifically uses the Joint Operation Planning and Execution System (JOPES) planning model and the Center-of-Gravity approach to operational design, but equivalent concepts could be used from alternative prescriptive models.

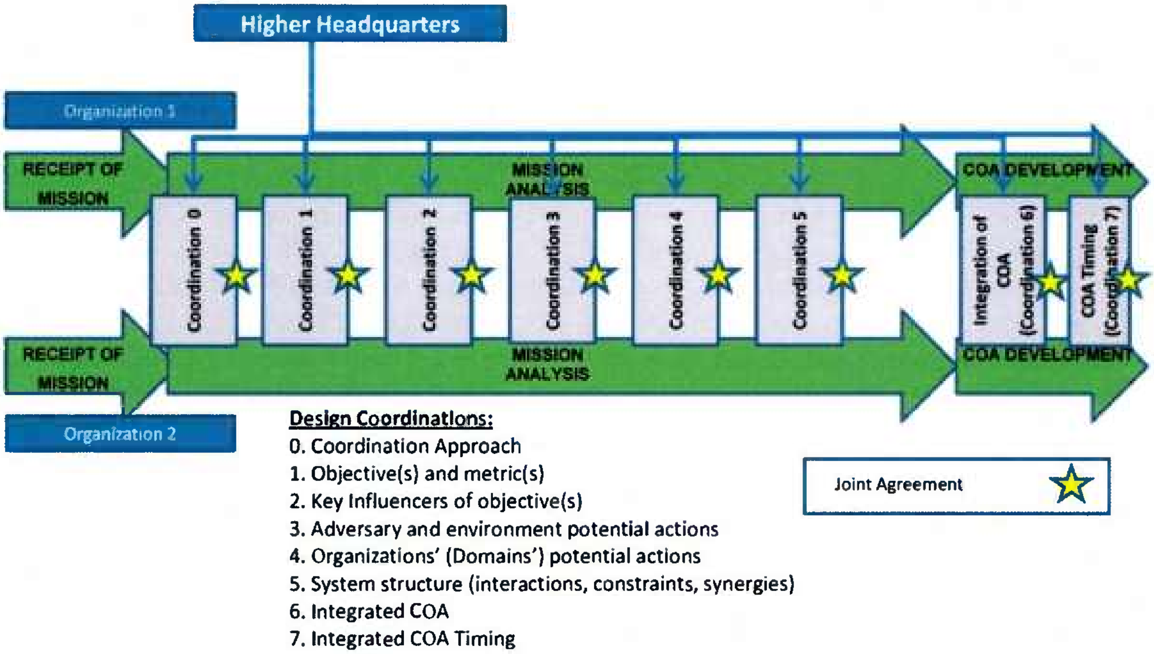


Fig. 27: Proposed Co-Design Approach

Using process activities from the current approach in the new approach to the extent possible also lowers the potential impact on total process time. The staff planning activities and commander design activities occur in the current approach. The additional activities of the proposed approach are the coordinations between commanders, or their designees, which occur concurrently with current approach activities. Although some additional process time is required to reach consensus on design coordinations, since the new activities are mainly concurrent with existing activities, the overall impact is less than traditional de-confliction activities. Traditional de-confliction activities take place after domain COA approval and are therefore not concurrent with other planning process activities. As a result, all the time required for de-confliction extends the overall process time required by an equal amount.

3.4 Modeling the Planning Process

Inter-domain coordination was modeled as iterative consensus building between domain decision makers. The five-stage interacting decision maker model was used as the basis for the iterative consensus building model [59]. The five stage interacting decision maker model builds upon classic decision making theory model of two stages, situation assessment and response selection [60], [61], by considering the additional stages for interacting with other decision makers and design support systems. In the situation assessment (SA) stage, decision makers create their assessment based on input from the environment or other decision makers. This assessment can be shared with other decision makers. Decision makers that receive shared information can fuse it during the information fusion (IF) stage. The fused information can be used in the task processing (TP) stage to select an approach to response selection (RS). The command interpretation (CI) stage accounts for restrictions to response selection place on decision makers by superior decision makers. In the final stage a response is selected which can be an organizational output or an input to another decision maker [59]. This model is shown in Fig. 28.

Table 6: Relationships among Planning Activity, Design Coordination, and Operational Design Elements

JOPES Activity	JOPES Output/Input to Design Coordination	Design Coordination	Output	Equivalent Doctrinal Design Concept
Determine Own & Enemy's Centers of Gravity and Critical Factors	Enemy Center of Gravity and Critical Factors	2. Key Influencers of objective(s)	Joint Key Influencers of Objectives	Critical Factors that Affect the Enemy Center of Gravity

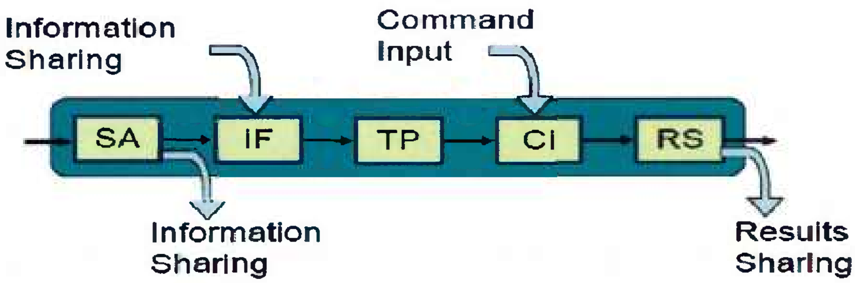


Fig. 28: The Five-stage Decision Maker Model

The five stage interacting decision maker model was extended to model iterative consensus building. Successive iterations were modeled by replicating the decision making organizations. These successive decision making organizations receive as input the results from that domain's previous decision and then during the information fusion stage gain understanding of the other domain's decisions and willingness to continue consensus building. In the response selection stage decision makers not only make a selection for the decision at hand but also determine whether they are willing to begin/continue consensus building. If any decision maker elects not to continue then the decisions will become final regardless of whether consensus has been obtained. Figure 29 demonstrates this process with two organizations and one iteration of consensus building. The coordination process structure is the same for all modeled coordination activity. The only exception is that the command interpretation stage is only used if there is appropriate command guidance. The number of iterations required to achieve full consensus for each type of coordination is a parameter examined in the subsequent analysis and can be deterministic or stochastic.

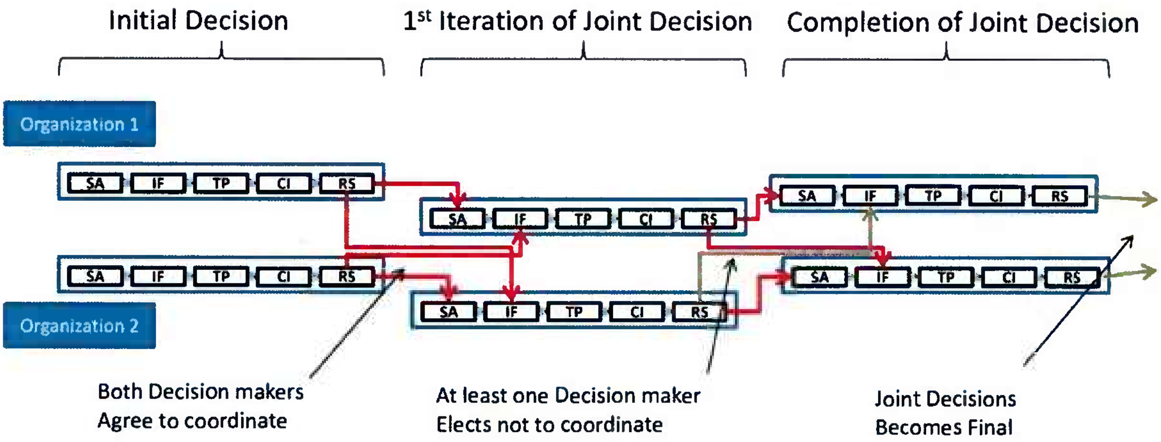


Fig. 29: The Iterative Consensus Building Modeling Approach

Conceptual modeling was accomplished using Timed Influence Nets (TINs) [62]. TINs were used to model both domain conceptual models and courses of action with performance estimates. The Pythia software tool was used to implement the TIN models used in this effort. Influence nets and timed influence nets are probabilistic belief networks with similarities to Bayesian Networks (BN). Unlike BN, TINs assume independence between casual influences which greatly simplifies the process of parameter elicitation by avoiding the requirement for eliciting extensive tables of conditional probability. The tables are instead constructed through the Causal Strengths (CAST) algorithm [63]. In situations where probability estimates are subjective, such as in strategic/operational course of action development, this assumption is appropriate. Previous research has demonstrated the effectiveness of TINs in operational and strategic level course of action development and modeling [64].

Based on a chosen scenario, one TIN was developed which represents the complete model of the operational environment. An example is shown in Figure 30. The performance of combined COAs will be measured using this complete model regardless of the approach used. COAs are chosen based on domain conceptual models, but the performance is based on applying those actions in the complete model. This complete model is the goal of conceptual model integration, representing a complete understanding of each domain's potential actions and their effects. Each domain will have this conceptual model on which to base COA selection if they conduct the proposed approach to build a common conceptual model. This complete model can be divided into eight types of nodes: actions for each of the three domains; goal node; key influencers of the goal node; standard enemy/environment effects; strong negative cross-domain effects; and strong positive cross-domain effects. The strong cross-domain effect nodes are designed to model the significant but non-obvious interactions that are not routinely discovered with the current approach.[3] The strong positive cross-domain effects are only discovered by creating a common conceptual model. Strong negative cross-domain effects can be identified through a common conceptual model or the more thorough level 2 de-confliction. During level 1 de-confliction, the

[3] The absence of discovery of these types of effects is evident in the continued emphasis on improved integration.

domains expand their domain-centric conceptual models to incorporate other domains' actions and effects. After successful completion of level 1 de-confliction, all domains have the same conceptual model encompassing all domain actions, goal node, key influencers of the goal node, and standard enemy/environment effects. At that point the domains can proceed to level 2 de-confliction, if they have chosen that approach.

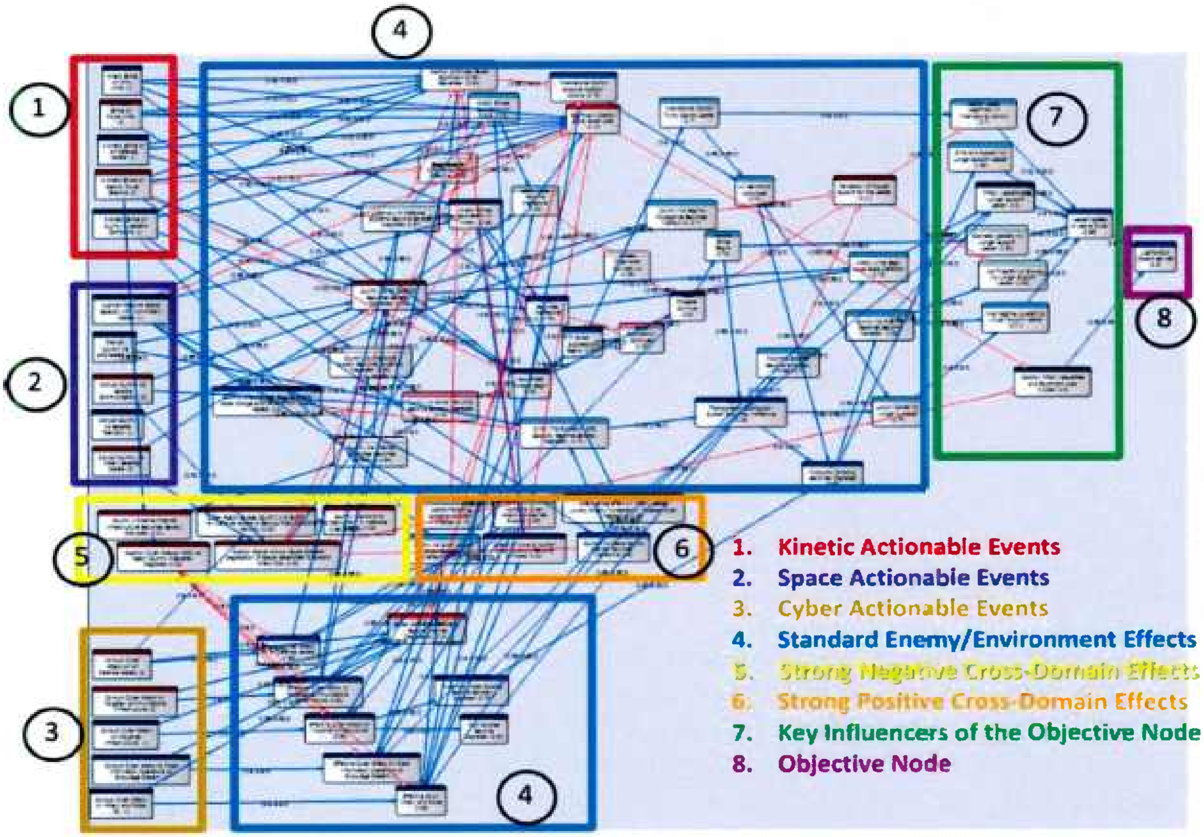


Fig. 30: The Complete Integrated Conceptual Model

Traditional domain centric conceptual models are also represented in a TIN. These domain models have the same goal node but are a subset of the complete TIN. These subsets are intended to model the knowledge of only the effects in the specific domain (and adversaries and neutral actors) without knowledge of the actions of adjacent domains; an example is shown in Fig. 31 for the kinetic domain. If no coordination is conducted, domains will choose COA based on respective domain model without any knowledge of the chosen actions of (or effects on) other domains.

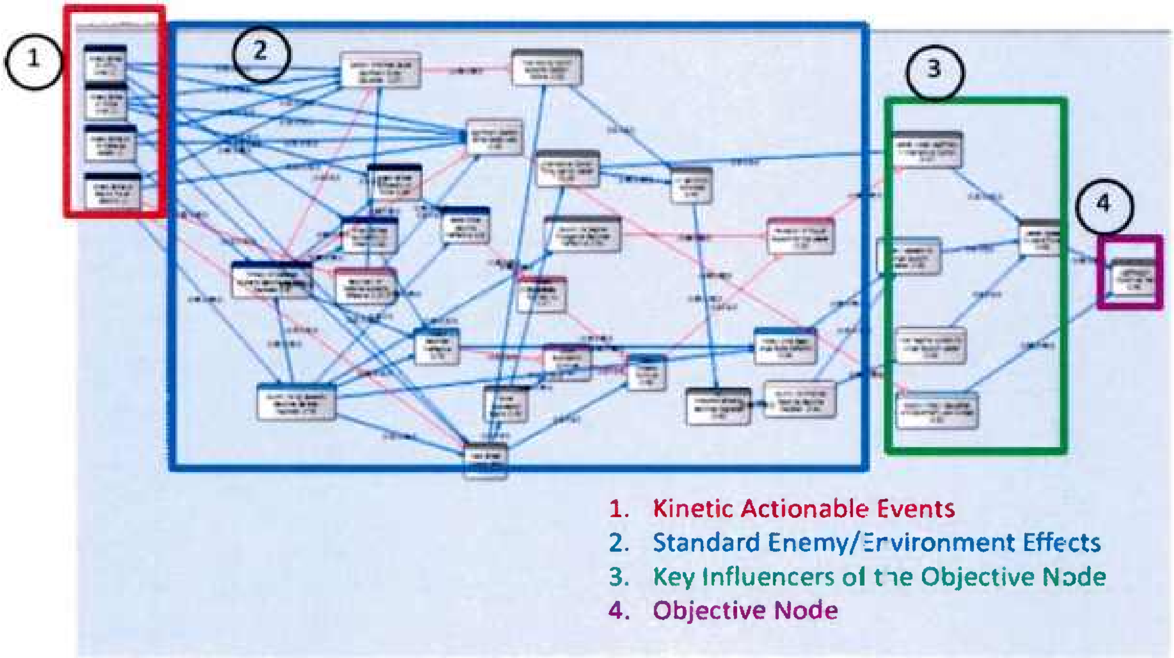


Fig.31: An Example Domain Conceptual Model

3.5 Experiment Results

The process model was limited to the military planning phases from receipt of mission to COA approval. The scenario chosen was a 48-hour time line for approval of an operational level COA. In the scenario, coalition forces have the goal of encouraging a brutal dictator to step down from power after he has lost international legitimacy. There is an equally weighted additional goal of preventing significant loss of coalition capability. Parameters considered during results analysis included: expected times for all planning activities, expected times for coordination activities, and expected number of coordination iterations to complete consensus building. In addition to the expected values for those parameters, each was also assigned a variance, zero for the deterministic case up to significant variance for different stochastic settings. The expected values were based on subject matter expert opinions from a current United States military command which conducts strategic and operational level planning. It would seem counter-intuitive that based on the planning activity time estimates the current approach including de-confliction can take longer than 48 hours when the time estimates are based on a 48 hour process. In other words, the apportionment of time from the 48 hours period for de-confliction is less than that required for full de-confliction. This is purposeful based on the feedback that, under current processes, full de-confliction is rarely achieved and time constraints often result in partially de-conflicted COAs. Another parameter examined was that of increasing time efficiency in subsequent consensus building iterations. As the leaders involved become increasingly familiar with the joint decision for which consensus is sought, it is possible that later iterations will take less time. This was modeled with two parameters: a percentage decrease by iteration in the original expected activity time and a minimum activity time.

Table 7 shows the deterministic results and Table 8 shows the stochastic results with significant variance on activity times and coordination iterations required. These results are based on the use of subject matter experts' opinions for parameter settings and computational experimentation with the described modeling methods. An exploration of the sensitivity of the various parameters has shown that these results are not particularly sensitive to any specific parameter value. Increasing the variability of the process times and iteration numbers increases the mean planning time (as would be expected in parallel processes) but the relative difference between the approaches remains fairly constant.

Table 7: Deterministic Model Results

Approach Used	Combined COA Type	Process Times (Without Iteration Efficiency) (CPN Model)		Process Times (With Iteration Efficiency) (CPN Model)		COA Performance (Probability of Goal Node Being True) (Pythia Model)		
		Minutes	Hours	Minutes	Hours	Coalition OBJs Met	Coalition Losses Avoided	Leader Agrees to Leave Power
New Approach	Integrated COA	3105	51.75	3007	50.11	0.802	0.9	0.85
Current Approach Level 2	De-conflicted Level 2	3385	56.42	2968	49.46	0.56	0.67	0.59
Current Approach	De-conflicted	3260	54.33	2860	47.66	0.394	0.45	0.43
No Coordination	Combined Domain COAs	2610	43.5	2610	43.5	0.28	0.32	0.295

Table 8: Stochastic Model Results

Approach Used	Combined COA Type	Process Times (Without Iteration Efficiency) (CPN Model)		Process Times (With Iteration Efficiency) (CPN Model)		COA Performance (Probability of Goal Node Being True) (Pythia Model)		
		Hours (Mean)	Hours (Std Dev)	Hours (Mean)	Hours (Std Dev)	Coalition OBJs Met	Coalition Losses Avoided	Leader Agrees to Leave Power
New Approach	Integrated COA	52.6	2.3	51.2	1.8	0.802	0.9	0.85
Current Approach Level 2	De-conflicted Level 2	57.4	1.3	50.2	1.1	0.56	0.67	0.59
Current Approach	De-conflicted	55	1.5	48.5	1.2	0.394	0.45	0.43
No Coordination	Combined Domain COAs	44	1.2	44	1.2	0.28	0.32	0.295

3.6 Conclusions

Based on estimates of realistic parameters for operational level COA development, the proposed approach provides significantly better integrated performance with at most a marginal increase (up to 5% depending on parameters) in the mean time required for the planning process. These results indicate the potential feasibility of the Co-design approach. However, there are several limitations in this approach to be addressed in further research described below. The approach articulates a framework for logical and efficient construction of a joint understanding of the operational environment between disparate domains. This work also demonstrates a new approach to the C2 planning process which emphasizes integrated planning and development of a common conceptual model. This is in contrast to most current approaches which simply increase sharing information with the expectation that integration will ensue without a specific supporting process. As a feasible alternative to current military planning approaches, Co-design offers an important design alternative for consideration in military command and control architectures.

A key assumption for the model approaches used is domain decision makers are properly motivated to come to consensus and will make choices which increase the likelihood of joint objective accomplishment. Research in many fields have shown the boundedness of rational decision making under various conditions [65]. It is also likely that in real military planning situations, domain leaders may have to balance competing domain objectives with common inter-domain objective(s). It is therefore important that experimentation with the Co-design approach be conducted with human decision makers with and without competing objectives. In addition, the focus of this effort has been on horizontal integration between domains; further research must be done on the application of Co-design within multiple levels of command. Another aspect to be explored is the effect on COA performance of compressing the time allowed for coordination processes in order to meet a strict planning timeline.

IV. USING MULTI-MODELING AND META-MODELING FOR C2

4.1 Introduction

Traditionally, Modeling and Simulation (M&S) environments were designed with the assumption that a single type of models would be developed and analyzed. A model in such an environment is developed using some known modeling techniques to address a certain class of problems. While single modeling techniques might be capable of answering specific questions, solving complex problems usually requires multiple models interoperating together (Multi-Modeling). The move towards supporting Multi-Modeling in various Modeling and Simulation platforms is already taking place. The Command and Control Wind Tunnel (C2WT) [66] developed by Vanderbilt University and the Service Oriented Architecture for Socio-Cultural Systems (SORASCS) [67] developed by Carnegie Mellon University are examples of Multi-Modeling capable platforms. While the first provides a federated approach, utilizing the High Level Architecture (HLA) [68] standard and the meta-programmable Generic Modeling Environment (GME) [69]; the second employs Service Oriented Architecture techniques in providing model interoperation capabilities.

In achieving Multi-Modeling, and to provide powerful supporting platforms, many challenges have to be faced. Beside the technical issues that usually arise in allowing interoperations between models through their modeling tools, there is also a major challenge of improving the human interface to the Multi-Modeling process itself [70]. This includes addressing both syntactic and semantic aspects of interoperation.

In this paper, a systematic methodology for addressing both syntactic and semantic issues in developing a Multi-Modeling approach to solve complex problems is presented. The focus of our approach is on helping users of Multi-Modeling platforms in designing workflows of Multi-Modeling activities that guarantee both syntactic and semantic correctness of the interoperations across models. Our approach is domain specific; the rationale behind this is twofold: first, problems to be solved by employing Multi-Modeling techniques are usually domain specific themselves; second, it narrows down the scope of meaningful interoperations among several modeling techniques where each technique offers unique insights and makes specific assumptions about the domain being modeled. We begin with identification and characterization of a domain of interest and its supporting modeling techniques. A Domain Analysis (DA) follows aiming to provide formal representations of syntactic and semantic aspects of the domain. A new Domain Specific Multi-Modeling Workflow Language is then developed to construct workflows that capture Multi-Modeling activities in the selected domain. A domain Ontology resulting from the Domain Analysis step is utilized to provide semantic guidance that effects valid model interoperation.

Our approach is illustrated in an application from the Drug Interdiction and Intelligence domain. The Joint Interagency Task Force - South (JIATF-South), an agency well known for interagency cooperation and intelligence fusion [71], receives huge amounts of disparate data regarding drug smuggling efforts. Analysis of such data using various modeling techniques is essential in identifying best Courses of Action (COAs). We apply our methodology to solve a class of problems in this domain by creating workflows of model interoperations involving Social Networks, Timed Influence Nets, Organization Structures, and Geospatial models.

In Section 4.2 we present a discussion of the basic concepts and approaches. The proposed methodology is presented in Section 4.3. Section 4.4 illustrates the application of the methodology. Conclusions and discussion of future work are in Section 4.5.

4.2 Multi-Modeling, Meta-Modeling and Workflows

Model interoperation has been addressed repeatedly, and from different perspectives, in the M&S research community. In this section we discuss some preliminary concepts and related approaches.

Modeling and Multi-Modeling: Modeling is the process of producing a model; a model is a representation of the construction and working of some situation of interest. [72] Figure 32a represents the modeling hierarchy where a Model is obtained using a Modeling Tool that applies a Modeling Language to represent a specific situation. The model itself should always conform to the Modeling Language used to create it. We call this combination of Model, Modeling Language and Modeling Tool a Modeling Technique.

We use multiple models because each Modeling Technique provides certain capabilities and makes specific assumptions about the domain being modeled. For example, Timed Influence Nets [73] describe cause and effect relationships among groups at high level but have no capability of capturing social aspects among the groups of interest. Social Networks [74], on the other hand, can describe the interactions among groups and members of the groups. In this context, a Multi-Modeling approach addresses a complex problem through the use of a number of interconnected domain-specific models where each model contributes insights to the overall problem. The interoperations between the interconnected models could serve different purposes and can happen in various forms.

Meta-Models and Meta-Modeling: A Meta-Model is an abstraction layer above the actual model and describes the Modeling Language used to create the model; the model has to conform to its Meta-Model. A Meta-Model conforms itself to a higher Meta-Model (Meta^2-Model) which describes the Meta-Modeling Language as shown in Fig. 32b.

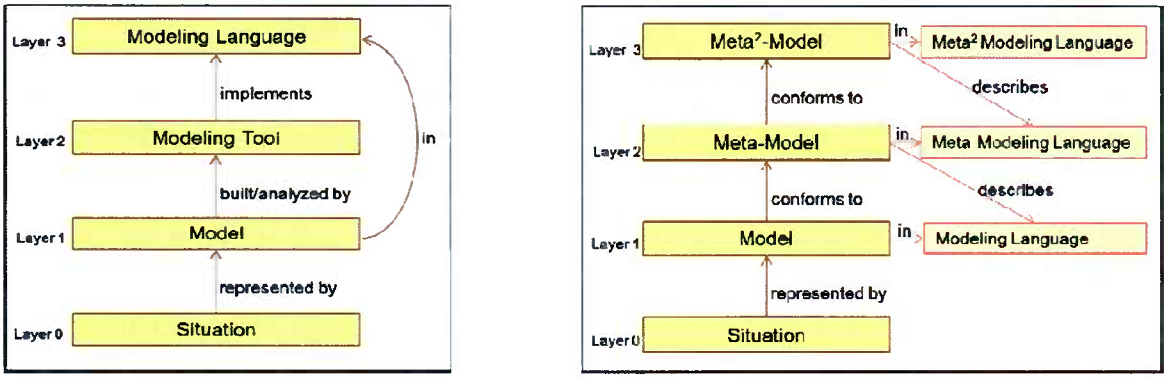


Fig. 32: (a) Modeling Hierarchy; (b) Meta-Models Hierarchy

The typical role of a Meta-Model is to define how model elements are instantiated. Meta-Modeling is defined to be the process of constructing a Meta-Model in order to model a specific problem within a certain domain. In the context of this Multi-Modeling research effort, the Meta-Modeling concept is extended to include the analysis of the conceptual foundations of a model ensemble. These models interoperate as part of a workflow developed to address a specific problem. Meta-Modeling then becomes a process of constructing a Meta-Model of a Multi-Modeling Workflow Language that captures interoperations between models.

Multi-Modeling Workflows: Four layers to be addressed in order to achieve Multi-Modeling. [75] The first layer, the Physical layer, i.e., Hardware and Software, is a platform that enables the concurrent execution of multiple models expressed in different modeling languages and provides the ability to exchange data and also to schedule the events across the different models. The second layer, the Syntactic layer, ascertains that the right data are exchanged among the models. The C2WT [66] and SORASCS [67] achieve that. In the third layer, the Semantic layer, interoperation across models should be examined to ensure that the exchange of data is semantically correct with respect to the problem domain. The fourth layer, the Workflow layer, is where workflows of interoperating models are captured.

A Multi-Modeling workflow is itself a model of an analysis process. A formal approach to capture a Multi-Modeling workflow requires a formal Modeling Language with its own rules. Developing workflows using such approach allows for translating visual views of model interoperation into an executable implementation. There already exist generic techniques for designing and implementing workflows such as Business Process Model and Notation (BPMN) [76] and Business Process Execution Language (BPEL) [77]. The domain specific nature of our approach requires us to develop a Domain Specific Multi-Modeling Workflow Language for the selected domain of interest. Such a language [78] would be tailored to a problem domain and would offer a high level of expressiveness and ease of use compared with a General Purpose Language (GPL) [79] and can be a specific profile of an existing GPL, i.e., BPMN. Figure 33 shows the mapping between our proposed Domain Specific Multi-Modeling Workflow Language (and its Meta-Model) to the Meta-Models Hierarchy.

Defining the Meta-Model of the workflow language in Layer 2 in Fig. 2 is a Meta-Modeling process itself. To capture those constructs of the Meta-Model that define the new language, a Meta-Modeling Language that conforms to a higher Meta-Model, Meta^2-Model, is also required. The research community in this area has addressed such hierarchy from different perspectives and many approaches were developed. One of these approaches is the Generic Modeling Environment (GME) [69], a configurable toolkit for creating domain-specific modeling languages and program synthesis environments, developed by Vanderbilt University. The configuration is accomplished through Meta-Models specifying the modeling paradigm (Modeling Language) of the application domain. The modeling paradigm contains all the syntactic and presentation information regarding the domain including the concepts used to construct models, relationships between concepts, different views and organizations of the concepts, and rules governing the modeling process. Defining the modeling paradigm is a modeling activity itself. GME has a Meta-Modeling paradigm that configures the environment for creating Meta-Models of the domain of interest. These models of the Meta-Models are then automatically translated into GME configuration through model interpretation. The Meta-Modeling paradigm is based on the Unified Modeling Language (UML).

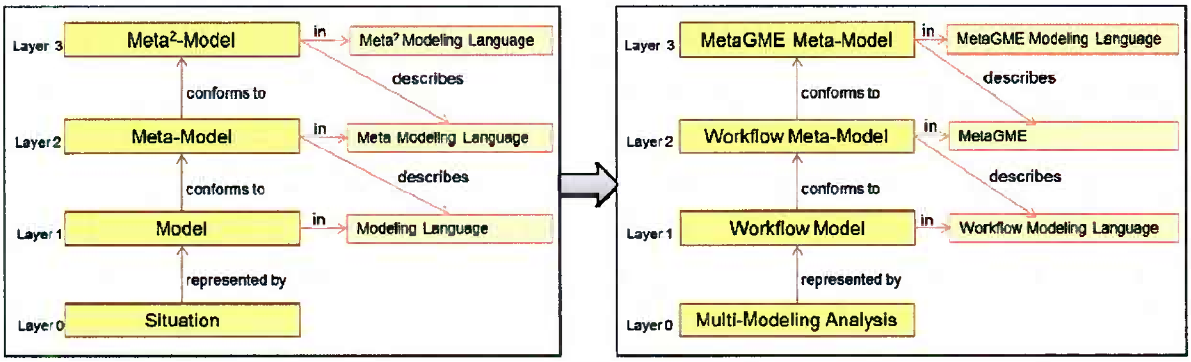


Fig. 33: Mapping Our Domain Specific Workflow Language to the Meta-Models Hierarchy

Ontologies: Ontology is the term used to refer to the shared understanding of a domain of interest. It entails some sort of a global view of a specific domain. This view is conceived as a set of concepts, their definitions and their inter-relationships; this view is referred to as conceptualization. In computer systems ontologies can be thought of as a means to structure a knowledge base. [80]

A Multi-Modeling Workflow Language needs a mechanism that guarantees semantic correctness of model interoperation. We propose the use of Ontology to guide interoperations between models. A Domain Specific Ontology that represents possible mappings between different concepts in the domain serves this purpose. The use of ontologies as a mean for representing the semantic aspects of model interoperability has been explored in [76] and [81]. The first approach is based on comparing ontologies (for each Modeling Technique) to help identify the similarities, overlaps, and/or mappings across the models under consideration and then constructing a higher level "Meta" ontology that determines which sets of models can interoperate. The second maintains a clear distinction between Meta-Models and Ontologies; they are different but complementary concepts, and both are needed to allow for model interoperation. We employ concepts from these two approaches in our methodology.

Domain Identification: Since the proposed Multi-Modeling approach for solving complex problems is domain specific, domain characterization becomes an essential task to be conducted prior to any other activity. The output of the domain characterization should provide enough information to perform domain analysis, construct domain ontology, and develop a Meta-Model of a Domain Specific Multi-Modeling Workflow.

In the context of software and systems engineering, a domain is most often understood as an applications area, a field for which systems are developed [82]. It is also defined to be a class of problems, where the types of problems to be solved and the context in which the system elements can be used are clearly identified [83]. In our approach, we consider a domain to be a specific class of problems to be solved using a set of Modeling Techniques and the appropriate required data.

The domain identification process itself has been approached in many research efforts, specially the research on software reusability in late 80's and early 90's. In [83] a comparison of Domain Analysis (DA) approaches for software reuse purposes was presented. Domain identification was pointed out as a first and essential step prior to any DA activities. Domain identification methods in those approaches include informal description in the form of statements, use of object oriented techniques, employing classification schemes, determining domain boundaries and collecting examples of similar problems.

4.3 The Multi-Modeling Approach

The focus of our approach is to provide a systematic methodology for creating and implementing Multi-Modeling Workflows that are both syntactically and semantically correct. It consists of five major steps as shown in Fig. 34. In this section we will discuss each step and its sub-steps in detail.

Domain Identification: This is the first step which deals with characterizing a specific domain of interest, in which, interoperating models are used to solve certain problems. We address the domain identification challenge by employing different techniques. As shown in Fig. 35a, we begin with an informal description of the domain in the form of statements that try to identify the problems to be solved, Modeling Techniques usually used in solving these problems, data sources and types, and main actors involved including domain experts, modelers and analysts.

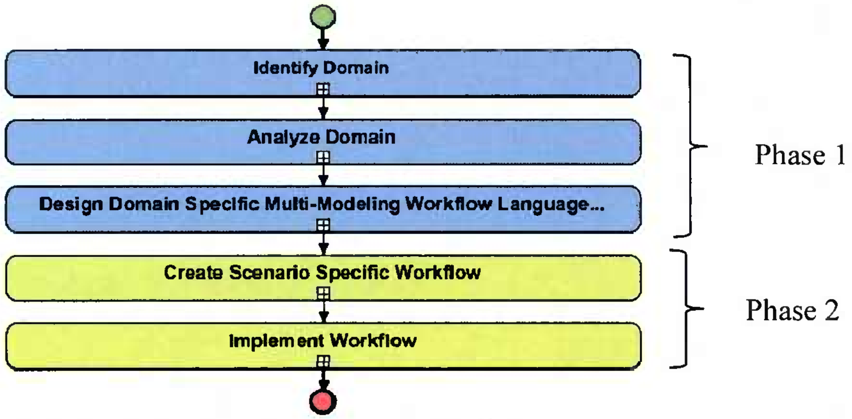


Fig. 34: Overview of the Proposed Methodology

After that we proceed with deciding the domain boundary in order to scope the domain and to exclude any unrelated elements. Then, a classification of concepts applicable to the domain takes place. These concepts serve as a repository for the final step. In the final step, Concept Maps [84] are used to represent those concepts. Concept mapping is a representation technique to organize knowledge about a specific domain. In our approach, we consider Concept Maps as a semi-formal representation of the domain. Generating Concept Maps is an iterative process until a satisfactory domain representation is reached.

Domain Analysis: Once satisfactory Concept Maps that represent the domain of interest and its supporting Modeling Techniques are ready, the Domain Analysis (DA) process starts. The process, as shown in Fig. 35b, goes into two parallel, but complementary, paths. On the outer path, UML class diagrams derived from the concept maps are produced to capture the structural aspects of the domain and its supporting Modeling Techniques. A mapping between these class diagrams follows to produce consolidated diagrams that include interoperations between the Modeling Techniques. On the inner path, ontologies based on the concept maps of the Domain and the Modeling Techniques are constructed to capture the semantic aspects. These ontologies are represented using the formal Web Ontology Language (OWL) standard. Mapping of these ontologies follows by employing Upper Ontology [85] and Ontology Matching [86] techniques.

Domain Specific Multi-Modeling Workflow Language: A Meta-Model of the new language has to be created and it should include the set of fundamental language constructs that represent the essential concepts of the domain, the set of valid relationships that exists between the domain concepts, and a set of constraints that govern how the language constructs can be combined to produce valid models. Accordingly, in the third step of our methodology, the UML class diagrams obtained from the DA step are used as the basis for the Meta-Model that defines the Domain Specific Multi-Modeling Workflow Language. The GME is used to create the Meta-Model of the Multi-Modeling Workflow Language. This Meta-Model is then automatically translated into a GME configuration that allows the use of GME itself to create workflows of specific Multi-Modeling scenarios. In general, we propose the use of a profile of BPMN as the basis of any Domain Specific Multi-Modeling Workflow Language.

Semantic Guidance of Multi-Modeling Workflows: The semantic concepts identified in the domain identification process and then captured in the Ontology in the domain analysis step should be enforced while using the new Domain Specific Multi-Modeling Workflow Language. Since our ontologies are represented in OWL [87] and we are using GME to create Multi-Modeling workflows, there should be a way to allow OWL ontologies to guide the creation of workflows; that is to guarantee their semantic correctness. GME allows for different types of extensions to the environment; basically using Plug-ins or Add-ons [69].

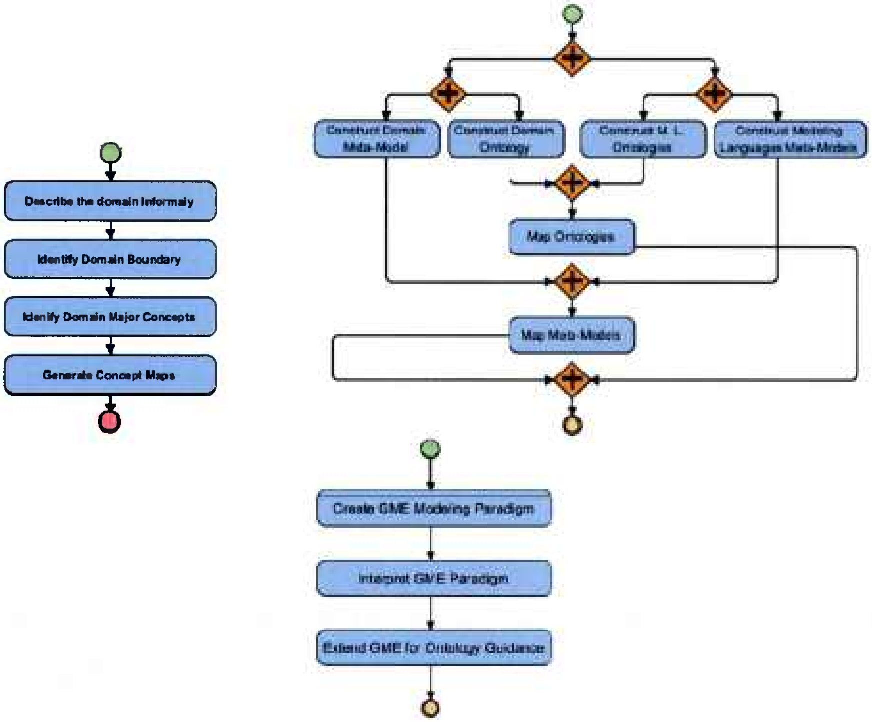


Fig. 35: (a) Domain Identification (b) Domain Analysis
(c) Multi-Modeling Workflow Language Definition

Utilizing these GME extensibility features and in order to address the semantic guidance issue, we implemented a GME Add-on extension. This extension reacts to GME events, and in case of any interoperation connection while using our Multi-Modeling Workflow Language, the OWL ontology is checked on the semantic validly of this connection. We use SPARQL [87] queries that are passed to a SPARQL Query Server to query the ontology. Based on the query result, our GME extension could allow or disallow the interoperation connection.

Multi-Modeling Workflow Creation: After defining the domain specific Multi-Modeling Workflow Language and having its GME modeling paradigm interpreted, GME can be used to create workflows for specific situations of interest. This is the fourth step of our methodology. Workflows constructed with our domain specific workflow language to capture interoperations across models are guaranteed to be both syntactically and semantically correct.

Workflow Implementation: The final step is to implement our workflows in an appropriate platform. In order to achieve this, an interpretation of the workflow to an executable form is required. For this purpose, a GME interpreter can be coded. One example platform that can be used to execute our workflows is the SORASCS [87]. SORASCS utilizes BPEL to execute workflows of analysis activities. Since we proposed basing our Domain Specific Multi-Modeling Workflow Language on BPMN, which can be mapped to BPEL [88], workflows created using our Multi-Modeling Workflow Language can be converted to executable workflows that SORASCS can execute.

Two Phase View of the Approach: The overall process of the methodology can be viewed as a two phase approach. Phase 1 is where the first three steps, domain identification, domain analysis, and workflow language definition take place. For a specific domain, this phase goes into multiple iterations until a Multi-Modeling Workflow Language that addresses a domain of interest and is capable of capturing model interoperations is reached. Phase 2 takes place when the Workflow Language is used to create workflows for specific scenarios. It is always possible to go back to Phase 1 to refine and enhance the Multi-Modeling Workflow Language; this might be the case when a new Modeling Technique is introduced in the domain of interest.

4.4 Application: JIATF-South

In this section we present an application of our approach to a decision making problem in the Drug Interdiction domain. The Joint Interagency Task Force - South (JIATF-South) is a Drug Interdiction agency well known for interagency cooperation and intelligence fusion [71]. The agency usually receives disparate data regarding drug trafficking and drug cartels from different sources. Quick and effective analysis of data is very essential in addressing drug trafficking threats effectively. A typical case begins with JIATF-South receiving information form the Drug Enforcement Administration (DEA). This prompts the deployment of Unmanned Airborne Vehicles (UAVs) that subsequently detect and monitor a suspect vessel until JIATF-South can sortie a Coast Guard cutter to intercept. If drugs are found, jurisdiction and disposition over the vessel, drugs and crew are coordinated with other agencies. Courses of Actions (COAs) identified by the agency are dependent on efficient analysis of received data.

In order to proceed in applying our approach, we first present a fictitious scenario of a possible drug trafficking activity reported to and monitored by JIATF-South. The scenario is presented briefly in Fig. 36.

- A US based drug cartel is involved in drug trafficking activity from Country R in the Caribbean to the USA.
- A cargo ship with R flag is being loaded with drugs. A drug cartel operating in country R is responsible.
- JIATF-South receives information about the drug smuggling activity from its intelligence sources in country R.
- The cargo ship disembarks the port of country R on Day x and goes under JIATF-South surveillance in international waters starting Day x+1. The ship is scheduled to arrive to the USA on day x+5.

Fig. 36: Scenario Brief

Analysts at JIATF-South are trained to use various Modeling Techniques to analyze data and then to identify possible COAs. In a traditional manner, analysts would be using each modeling technique individually. By applying our methodology, an efficient Multi-Modeling based analysis would make such analysis process more accurate and faster. The rationale is that while each Modeling Technique might be capable of capturing certain aspects of the available data, interoperation between models will definitely improve the results of the overall process. Also, the ability for analysts to create visual workflows of the Multi-Modeling activity provides a mean of reusability of the constructed workflows in addressing similar scenarios. In this section we show a practical example of applying our 5-steps methodology to the JIATF-South drug interdiction scenario.

Domain Identification: We first begin with an informal description of the domain. Looking back at the JIATF-South operations description and the brief scenario, the following partial list of statements shown in Fig. 37 describes the main concepts of the domain.

- Drug Interdiction involves information sharing, fusion of intelligence data and monitoring of drug trafficking activities.
- Given (incomplete and uncertain) information, decisions to be made on best COAs.
- Drug Interdiction involves dealing with Drug Cartels and Smugglers (RED groups) and Law Enforcement and Intelligence (Blue groups).
- Drug Smuggling takes different routes and originates from different sources.
- Analysts use Social Networks, GIS, Influence Nets, Asset Allocation and Scheduling and Organization Models techniques.

Fig. 37: Informal Description of Domain

These informal statements are then revised to scope the domain and exclude any concepts that are outside its boundary. In this example, the Asset Allocation and Scheduling problem is not addressed. A repository of related concepts is then identified. Table 9 shows examples of some related concepts. The concepts are classified into two major categories, Domain Concepts, and Modeling Techniques Concepts.

Table 9. Domain and Modeling Techniques Concepts

General Domain Concepts	Specific Domain Concepts	Modeling Techniques Concepts	Specific M. Techniques Concepts
Drug Interdiction	Drug Smuggling Drug Cartels	Geospatial Analyses	Incidents, Time Location, Route WebTas
Data	Geospatial Time Individuals	Influence Nets	Node Link Proposition Probability
Interagency Collaboration	Intelligence Agencies Law Enforcement Agencies		

After identifying related concepts, we construct Concept Maps to capture the relations between these concepts. Concept Maps are generally constructed to answer specific questions in the domain of interest. Figure 38 shows a concept map that addresses the question: How does JIATF-South perform Drug Interdiction? The same applies to other aspects of the domain and the Modeling Techniques. As part of the process, we refer to similar experiences and make use of existing assets. In [76], concept maps for Influence Nets and Social Networks were constructed.

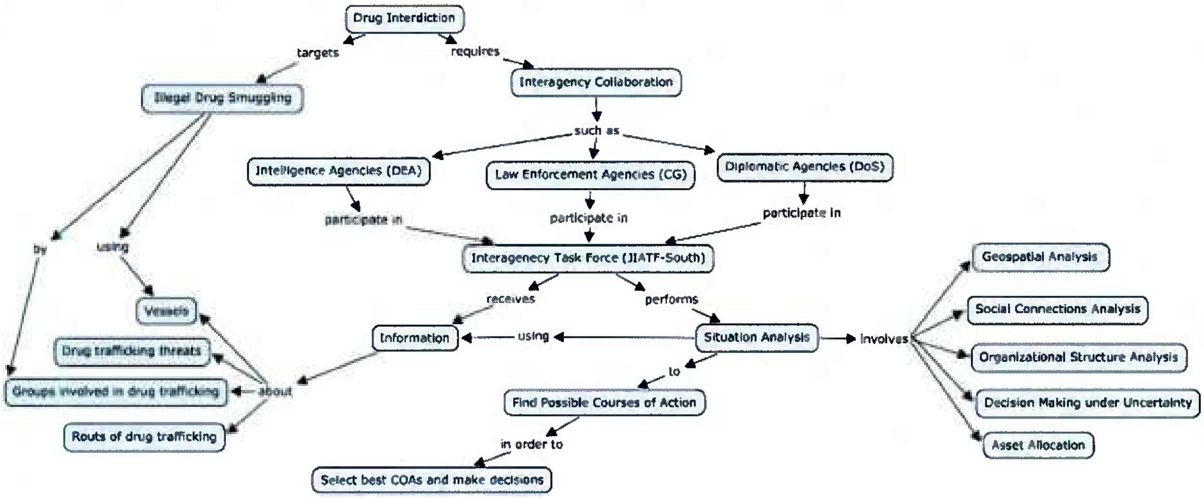


Fig. 38: Concept Map: How does JIATF-South perform Drug Interdiction?

Domain Analysis: In this step, we use the generated Concept Map to perform domain analysis. We construct UML class diagrams to represent the constructs of the domain and the Modeling Techniques. Parallel to that, we identify semantic concepts and relations and capture them in our OWL Ontology. In Fig. 39, we show a partial UML class diagram that represents the constructs of the drug interdiction domain.

Domain Specific Multi-Modeling Workflow Language: Using GME, a Meta-Model for our domain's Workflow Multi-Modeling Language is defined. This Meta-Model defines the constructs of this new language. In addition to basic constructs borrowed from BPMN, we have introduced some new constructs and imposed some constraints. A workflow in our domain has two types of activities, operations and interoperations. Operations are those activates performed on a specific model using the modeling tool that supports its modeling language. Interoperations are those activities that involve interoperations across models through their modeling tools. Operations in our language can be in one of two flavors, Thick or Thin Operations. This is due to the fact that Multi-Modeling platforms can support the integration of Modeling Tools in one of two forms. Thin Operations represent the case when service based integration takes place, given that the modeling tool of interest exposes its functionalities as services. Thick Operations represent the case in which the whole Modeling tool is integrated as a package in the Multi-Modeling platform.

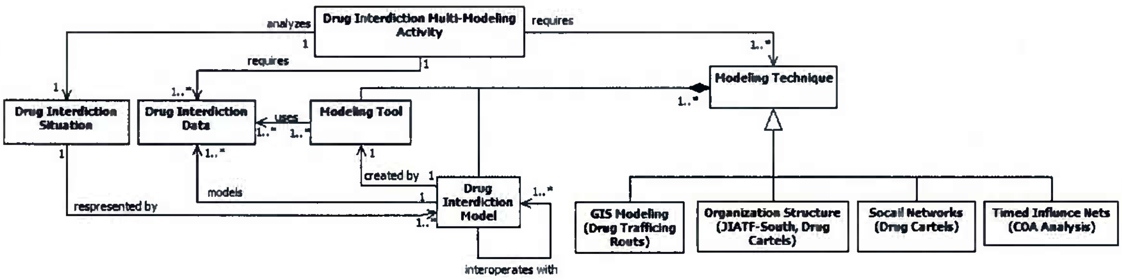


Fig. 39: UML Class diagram representing main constructs of the drug interdiction domain

Multi-Modeling Workflow Creation: Once the GME Meta-Model of our Domain Specific Multi-Modeling Workflow Language is interpreted and registered as a new Modeling Paradigm in GME, we begin using the GME environment to create workflows that capture specific domain scenarios. In Fig. 40 we show a workflow that involves the use of GIS, Timed Influence Nets, Social Networks and Organization Models to analyze data and then generate and select best COAs.

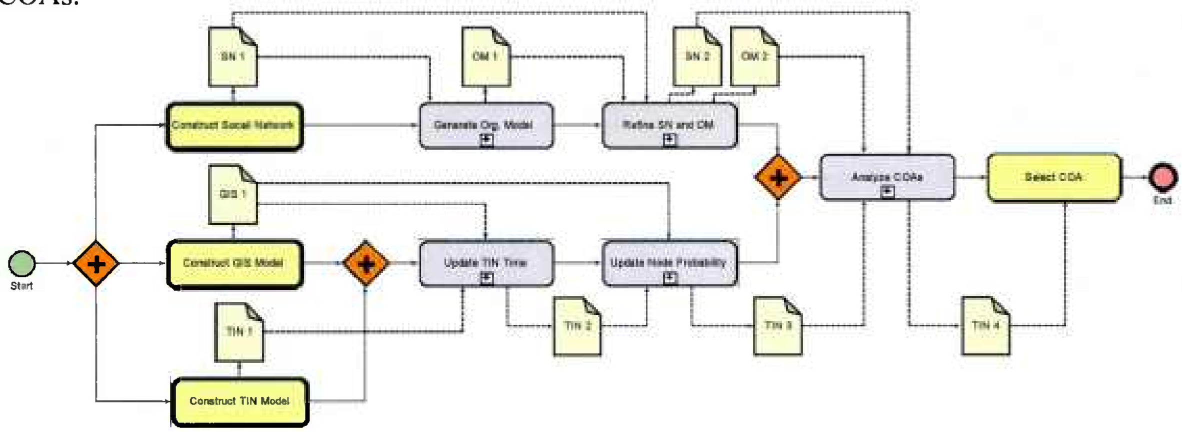


Fig. 40: Workflow of a Drug Interdiction Multi-Modeling Activity

4.6 Conclusion

A systematic methodology for addressing Multi-Modeling problems by employing a Domain Specific Multi-Modeling Workflow Language and a supporting Domain Ontology has been developed. Our approach is domain specific and requires the characterization of a specific domain, modeling techniques used in solving problems in that domain, and data sources available for creating models of specific scenarios in that domain. Domain characterization is a complex problem by itself. It has been addressed in our proposed methodology by building on top of previous research in this area. We believe that this is an area that deserves more future attention as it is essential for making our approach capable of capturing different possible combinations of Multi-Modeling activities for a specific domain.

V. REFERENCES

[1] INCOSE, Resilient Systems Working Group homepage: http://www.incose.org/practice/techactivities/wg/rswg/

[2] S. Jackson, *Architecting Resilient Systems: accident avoidance and survival and recovery from disruptions.* Hoboken, NJ: Wiley Series in Systems Engineering and Management, 2010.

[3] ISO/IEC Systems and software engineering - *Recommended practice for architectural description of software-intensive systems*, ISO/IEC 42010, 2007

[4] P. H. Cothier and A.H. Levis, *Timeliness and Measures of Effectiveness in Command and Control*, IEEE Transactions on Systems, Man, and Cybernetics Vol SMC-16, No 6, November/December 1986.

[5] V. Bouthonnier and A.H. Levis, *Effectiveness of C^3 Systems*, IEEE Transactions on Systems, Man, and Cybernetics Vol SMC-16, No 14, January / February 1984

[6] M. A. Pflanz "*On the Resilience of Command and Control Architectures*" PhD Dissertation, Volgenau School of Engineering, George Mason University, Fairfax, VA, November 2011.

[7] F.R.H. Valraud and A. H. Levis, *On the Quantitative Evaluation of Functionality in C3 Systems*, AFCEA International Press, pp132-139, August 1989.

[8] J. Martinez and M. Silva, C. Girrault and W. Reisig, "*A simple and fast algorithm to obtain all invariants of generalized Petri nets*", Informatik-Fachbrichte 52, pp. 301 - 303 , 1982: Springer-Verlag

[9] S.W. Liles "*On The Characterization and Analysis of System of Systems Architectures*" PhD Dissertation, Dept of Information Technology and Engineering, George Mason University, Fairfax, VA, August 2008

[10] Guide to the Systems Engineering Body of Knowledge (SEBoK), *"Resilience Engineering,"* Available: http://www.sebokwiki.org/index.php/Resilience_Engineering

[11] US Navy Public Affairs, 2009. US 4th Fleet, Specialist 2nd Class Alan Gragg, *U.S. 4th Fleet Stands Up Maritime Operations Center*, 26 March 2009, [Online] Available at: http://www.navy.mil/search/display.asp?story_id=43775 [Accessed August 2011].

[12] Remy, P. A., Jin, V. Y., & Levis, A. H., 1988. On the Design of Distributed Organization Structures, *Automatica*, 24(1).

[13] Kansal, S.K., AbuSharekh, A.M., & Levis, 2007. A.H., Computationally Derived Models of Adversary Organizations, In: *Proc. IEEE Symp. on Computational Intelligence for Security and Defense Applications,* April 2007, Honolulu, HI.

[14] Liles, S., 2008. *On The Characterization and Analysis of System of Systems Architectures* PhD Dissertation, Dept. of Information Technology and Engineering, George Mason University, Fairfax, VA, August 2008.

[15] J. K. DeRosa and M. D. Woodall, "Evolution To Integrated Command and Control," NASA, 1998.

[16] D. S. Alberts and R. Hayes, Power to the Edge: Command and Control in the Information Age. 2003.

[17] Chief of the Defence Staff, "Canadian Forces Joint Publication 5.0." Apr-2008.

[18] Department of Defense, "Command and Control Joint Integrating Concept." 01-Sep-2005.

[19] Chairman of the Joint Chiefs of Staff, "Joint Operations." 2010.

[20] Department of Defense, "Quadrennial Defense Review Report," Feb. 2010.

[21] J. A. St Laurent, "Military Operations. Actions Needed to Improve DOD's Stability Operations Approach and Enhance Interagency Planning," DTIC Document, 2007.

[22] P. Louvieris, N. Mashanovich, C. Collins, G. White, M. Faulkner, J. Levine, S. Henderson, and D. T. C. Surrey, "Exploring Joint Usability and Decision Effectiveness using a Networked-Enabled Virtual Collaborative Working and Visualisation Environment for Military Planning," 2008.

[23] D. A. Gilmour, "Real-Time Course of Action Analysis," Information Directorate, Air Force Research Lab, Rome, 2006.

[24] Q. Huang, J. Haallmats, K. Wallenius, and J. Brynielsson, "Simulation-based decision support for command and control in joint operations," in Proceedings of the 2003 European Simulation Interoperability Workshop, p. 091.

[25] T. Clark and T. Moon, "Interoperability for Joint and Coalition Operations," Australian Defence Force Journal, vol. 151, no. November/December, pp. 23–36, 2001.

[26] C. Zins, "Conceptual approaches for defining data, information, and knowledge," J. Am. Soc. Inf. Sci., vol. 58, no. 4, pp. 479–493, Feb. 2007.

[27] S. G. McIntyre, M. Gauvin, and B. Waruszynski, "Knowledge management in the military context," Canadian Military Journal, vol. 4, no. 1, pp. 35–40, 2003.

[28] W. Perry, Information Sharing Among Military Headquarters: The Effects on Decisionmaking. 2004.

[29] D. J. Bryant, "Can We Streamline Operational Planning?," Canadian Military Journal, vol. 7, no. 4, pp. 2006–2007, 2007.

[30] J. Vowell, "Between Discipline and Intuition: The Military Decision Making Process in the Army's Future Force," US Army School for Advanced Military Studies,250 Gibbon Ave, Fort Leavenworth,KS,66027, May 2004.

[31] N. R. McCown, "Developing Intuitive Decision-Making In Modern Military Leadership," Operations Department, Naval War College, Newport, RI, 2010.

[32] C. R. Paparone, "US Army Decision Making: Past, Present and Future," Military Review, Aug. 2001.

[33] M. Vego, "Effects-Based Warfare: A Critical View."

[34] Z. Jobbagy, "Literature Survey on Effects-Based Operations," Aug. 2003.

[35] E. A. Smith, Effects Based Operations: Applying Network Centric Warfare in Peace, Crisis, and War. .

[36] M. J. Hannan, "Operational Net Assessment: A Framework for Social Network Analysis," IO Sphere.

[37] C. H. Canon, "Systemic Operational Design: An Alternative to Estimate Planning," Operations Department, Naval War College, Newport, RI, 2009.

[38] C. D. Dalton, "Systemic Operational Design: Epistemological Bumpf or the Way Ahead for Operational Design?," 2006.

[39] W. T. Sorrells, G. R. Downing, P. J. Blakesley, D. W. Pendall, J. K. Walk, and R. D. Wallwork, "Systemic Operational Design: An Introduction," School of Advanced Military Studies United States Army Command and General Staff College Fort Leavenworth, Kansas, 2005.

[40] Headquarters Department of the Army, The Operations Process (FM 5-0). Headquarters Department of the Army, 2010.

[41] Chairman of the Joint Chiefs of Staff, "Joint Operation Planning." 26-Dec-2006.

[41] Chairman of the Joint Chiefs of Staff, "Joint Operation Planning and Execution System (JOPES) Volume I (Planning Policies and Procedures)." Joint Staff Washington, D.C. 20318, 2001.

[43] J. D. Thompson, Organizations in Action. McGraw-Hill, 1967.

[44] M. Mullen, "The National Military Strategy of the United States." Joint Chiefs of Staff, 2011.

[45] "Organizational Transformation: Military Departments Can Improve Their Enterprise Architecture Programs." Government Accountability Office, 2011.

[46] T. F. Jenkins, A. D. Hewitt, C. L. Grant, S. Thiboutot, G. Ampleman, M. E. Walsh, T. A. Ranney, C. A. Ramsey, A. J. Palazzo, and J. C. Pennington, "Description and Analysis of Military Planning Systems," 2006.

[47] S. Liu, Duffy, A.H.B., and Whitfield, R.I, "Towards the Realisation of an Integratated Decision Support Environment for Organisational Decision Making," International Journal of the Decision Support System Technology, vol. 1, no. 4, pp. 35–58, 2009.

[48] M. Chmielewski, "Ontology Applications for Achieving Situation Awareness in Military Decision Support Systems," Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, pp. 528–539, 2009.

[49] J. Hanna, "Course of Action Simulation Analysis," Science Applications International Corp (SAIC) Beavercreek Ohio, 2005.

[50] G. E. Seymour and M. B. Cowen, "A Review of Team Collaboration Tools Used In the Military and Government," 2007.

[51] P. Louvieris, A. Gregoriades, and W. Garn, "Assessing critical success factors for military decision support," Expert Systems with Applications, vol. 37, no. 12, pp. 8229–8241,2010.

[52] "Defense Acquisitions: Steps Needed to Ensure Interoperability of Systems That Process Intelligence Data." United States General Accounting Office, 2003.

[53] S. C.-Y. Lu and J. Cai, "A Collaborative Design Process Model in the Sociotechnical Engineering Design Framework," Artif. Intell. Eng. Des. Anal. Manuf., vol. 15, no. 1, pp. 3–20, Jan. 2001.

[54] K. Schmidt, "Modes and Mechanisms of Interaction in Cooperative Work," RisU00F8 National Laboratory, Roskilde, Denmark, 1994.

[55] H.-Z. Huang, H.-W. Xu, and Xu Zu, "Petri Net-based Coordination Component for Collaborative Design," Concurrent Engineering, vol. 18, no. 3, pp. 199–205, Sep. 2010.

[56] G. L. Klein, J. L. Drury, M. Pfaff, and L. More, "COA Action: Enabling Collaborative Option Awareness," 2010.

[57] G. L. Klein and L. Adelman, "A Collaboration Evaluation Framework," in Proceedings of the 2005 International Conference on Intelligence Analysis, 2005.

[58] T. Saltysiak, "On Approaches for Integrated Course of Action Development," PhD Disseration, Dept. of Systems Engineering and Operations Research, George Mason University, Fairfax, VA, 2012.

[59] A. H. Levis, "A Colored Petri Net Model of Command and Control Nodes," in Toward a Science of Command Control and Communications, Carl R. Jones., Washington, DC: AIAA Press, 1993.

[60] J. G. March and H. A. Simon, Organizations. Oxford, England: Wiley, 1958.

[61] H. Mintzberg, D. Raisinghani, and A. Théorêt, "The Structure of 'Unstructured' Decision Processes," Administrative Science Quarterly, vol. 21, no. 2, pp. 246–275, Jun. 1976.

[62] L. W. Wagenhals, I. Shin, and A. H. Levis, "Creating Executable Models of Influence Nets with Colored Petri Nets," International Journal on Software Tools for Technology Transfer (STTT), vol. 2, no. 2, pp. 168–181, Dec. 1998.

[63] K. C. Chang, P. E. Lehner, A. H. Levis, A. K. Zaidi, and Z. Xinhai, "On Causal Influence Logic," George Mason University, Center of Excellence for C3I, 1994.

[64] L. W. Wagenhals, T. Reid, R. J. Smillie, and A. H. Levis, "Course of Action Analysis for Coalition Operations." Jun-2001.

[65] H. A. Simon, G. B. Dantzig, R. Hogarth, C. R. Plott, H. Raiffa, T. C. Schelling, K. A. Shepsle, R. Thaler, A. Tversky, and S. Winter, "Decision making and problem solving," Interfaces, pp. 11–31, 1987.

[66] G. Hemingway, H. Neema, H. Nine, J. Sztipanovits, and G. Karsai, Rapid Synthesis of High-level Architecture-Based Heterogeneous Simulation: a Model-Based Integration Approach, Simulation, (2011).

[67] D. Garlan, K. M. Carley, B. Schmerl, M. Bigrigg, and O. Celiku, Using Service-Oriented Archi-tectures for Socio-Cultural Analysis, Software Engineering and Knowledge Engineering, (2009).

[68] HLA Evolved Working Group. (2010) IEEE Standards Association. [Online]. http://standards.ieee.org/findstds/standard/1516-2010.html

[69] James Davis, GME: The Generic Modeling Environment, Conference on Object-oriented programming, systems, languages, and applications, (2003).

[70] Paul A. Fishwick, Toward an Integrative Multimodeling Interface: A Human-Computer Interface Approach to Interrelating Model Structures, Simulation, (2004).

[71] Evan Munsing and Christopher Lamb, Joint Interagency Task Force - South: The Best Known, Least Understood Interagency Success, 2011.

[72] A. Maria, Introduction to Modeling and Simulation, Proceedings of the 29th Winter Simulation Conference, (1997).

[73] S. Haider and A. H. Levis, Effective Course-of-Action Determination to Achieve Desired Effects, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, vol. 37, no. 6, pp. 1140 - 1150 , (2007).

[74] K. Carley, On the Evolution of Social and Organizational Networks, Steven B. Andrews and David Knoke (Eds.) special issue of Research in the Sociology of Organizations, vol. 16, pp. 3-30, (1999).

[75] Alexander H. Levis, Abbas K. Zaidi, and Mohammad F. Rafi, Multi-modeling and Meta-modeling of Human Organizations, 4th International Conference on Applied Human Factors and Ergonomics, (2012).

[76] OMG (2011) Business Process Model and Notation. [Online].

http://www.omg.org/spec/BPMN/

[77] OASIS. (2007) Business Process Execution Language. [Online].

https://www.oasis-open.org/committees/wsbpel/

[78] M. Mernik, J. Heering, and A. M. Sloane, When and how to develop domain-specific languages, ACM Computing Surveys (CSUR), vol. 37, pp. 316–344, (2005).

[79] M. Uschold, M. Gruninger, M. Uschold, and and M. Gruninger, Ontologies: Principles, Methods and Applications, Knowledge Engineering Review, vol. 11, pp. 93-136, (1996).

[80] P. Höfferer, Achieving business process model interoperability using meta-models and ontologies, 15th European Conference on Information Systems, (2007).

[81] Ruben Prieto-Diaz, Domain Analysis: An Introduction, Software Engineering Notes, vol. 15, no. 2, p. 47, (1990).

[82] X. Ferré and S. Vegas, An Evaluation of Domain Analysis Methods, 4th International Workshop in Evaluation of Modeling Methods in Systems Analysis and Design, (1999).

[83] Joseph D. Novak and Alberto J. Cañas, The Theory Underlying Concept Maps and How to Construct and Use Them, 2006,

[84] I. Niles and A. Pease, Towards a standard upper ontology, International conference on Formal Ontology in Information Systems, (2001).

[85] K. Kotis and M. Lanzenberger, Ontology Matching: Current Status, Dilemmas and Future Challenges, International Conference on Complex, Intelligent and Software Intensive Systems, (2008).

[86] Sean Bechhofer et al. OWL Web Ontology Language. [Online].

http://www.w3.org/TR/owl-ref/

[87] W3C. (2008) SPARQL Query Language for RDF. [Online].

http://www.w3.org/TR/rdf-sparql-query/

[88] C. Ouvans, From BPMN Process Models to BPEL Web Services, International Conference on Web Services, (2006).